Balancing Multiple Objectives in Slate Recommendation Using Reinforcement Learning

Yassin Ben Allal[1]
Student number: 2602029
First supervisor: Dr. Floris den Hengst[1]
Second reader: Dr. Frank van Harmelen[1]

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

Abstract. Recommender systems often focus solely on optimizing user engagement, which can lead to undesirable side-effects such as filter bubbles and lack of content diversity. While multi-objective optimization could help balance these competing goals, its application to slate recommendation — where multiple items are recommended simultaneously — remains unexplored. This thesis investigates the use of multi-objective reinforcement learning (MORL) to simultaneously optimize engagement and diversity in slate recommendation. We extend the SARDINE simulator to support multi-objective evaluation, and implement a novel combination of Prediction-Guided MORL (PGMORL) with SAC+TopK for slate construction. Through extensive experiments across different catalog sizes (100-1682 items) and slate sizes (3-20 items), we demonstrate that our approach can effectively discover policies representing different trade-offs between engagement and diversity. Results show that PGMORL consistently finds Pareto-optimal policies that dominate single-objective approaches, though at increased computational cost. While our method requires more resources than single-objective alternatives, its runtime scales reasonably with slate size, making it a practical solution for real-world recommendation scenarios. This work provides important insights into the feasibility and challenges of multi-objective optimization in slate recommendation systems.

Keywords: Slate Recommendation · Multi-Objective Reinforcement Learning · Recommender Systems.

1 Introduction

Recommender systems (RSs) personalize user experiences across platforms, such as Netflix and Amazon, by filtering choices to align with user interests. While effective, engagement-focused RSs may limit content diversity. Solely optimizing for engagement could have undesirable side-effects, e.g., espousing click-bait content [21].

Enhancing the diversification of personalized recommendations, which caters to a user's wider range of interests.[42, 30, 5, 35, 21]. Furthermore, recommendation diversity plays an important role in increasing user-satisfaction in the long-term, improving user experience, and it helps prevent the occurrence of myopic recommendations [2, 25, 33]. Diversity is not the only feature of a recommendation that facilitates desired outcomes for a RS; novelty is important as well [22]. Both metrics are fundamental metrics for RSs that go beyond engagement [4], sometimes even promoting enhancement in the click rate [13].

Thus, despite an initial decrease in engagement, diversity improves the user satisfaction over the long term. This increase in user satisfaction can be explained using an example. Consider a user who frequently listens to classic rock bands on Spotify. A recommender system optimizing solely for engagement will suggest similar bands, missing potential interests in other genres. While recommending more classic rock might maximize immediate engagement, it fails to account for the user's unexpressed interests in musical exploration. As this simple example demonstrates, recommender algorithms operate on incomplete knowledge, basing their decision on a limited sample of user activity. Due to this uncertainty in the user's true interests, diversity and novelty are therefore good strategies to help optimize the accuracy of a RS. This inherent desire in diversity and novelty is furthermore confirmed by various consumer behavior studies highlighting the variety seeking drive in human behavior [27, 32]. Novel and diverse recommendations enhance the user experience by expanding their horizons and fostering new interests [4].

Retaining diversity in recommendations is essential for long-term user engagement and satisfaction. This recommendation problem can be approached as a sequential decision-making process, modeled by a Markov Decision Process [36]. Sequential decision problems can be modeled as a Markov Decision Process (MDP). An MDP provides a framework for sequential decision-making with states, actions, transition probabilities, and rewards, allowing an agent to maximize cumulative rewards over time [31]. In recommender systems, an action is a recommended item, the state includes information about the environment (e.g., a user's click history), and the reward signal reflects user behavior, which we aim to optimize for sustained engagement. Reinforcement Learning (RL) algorithms are effective for maximizing these long-term rewards.

In many Reinforcement Learning-based Recommender Systems (RLRSs), actions are defined as a single item to be recommended. However, in practice, it makes more sense to recommend a list or slate of items to a user simultaneously, as illustrated in figure 1. Slate recommendation gives humans the opportunity to be part of the decision process.

In the field of slate recommendation, previous research has explored various approaches to improve recommendation quality. While some methods implicitly consider multiple aspects of recommendation quality [23], no research has yet explicitly optimized multiple objectives simultaneously using slate RLRSs. This is particularly relevant for slate recommendation, as recommending multiple items simultaneously provides natural opportunities to optimize for objectives like diversity—the variety of items within a slate can be directly measured and optimized. Different combinations of items may lead to different short and long-term outcomes, making the explicit optimization of multiple objectives par-

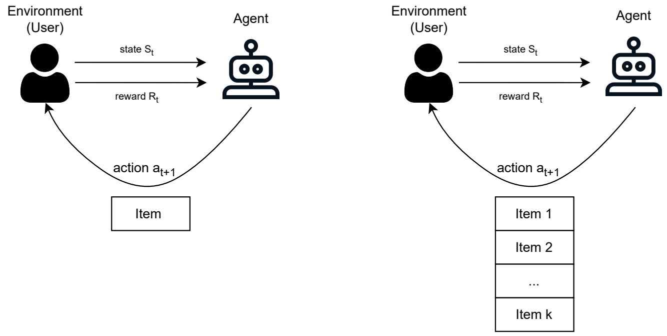


Fig. 1: Diagram showing the difference between regular RLRS and slate RLRS[1]

ticularly promising for improving overall recommendation performance. In this work, we present the first implementation of explicit multi-objective optimization for slate recommendation using RL, allowing for direct trade-offs between engagement and diversity objectives. We extend the SARDINE simulator [12] to support multi-objective evaluation, enabling comprehensive testing of trade-offs between engagement and diversity in slate recommendation. Through empirical evaluation, we demonstrate the effectiveness of our approach in balancing these competing objectives. Now is the right time to develop a feasible multi-objective approach system for slate recommenders. Therefore, in this paper we propose to answer the following research question:

Research Question

How can multi-objective reinforcement learning be used to simultaneously optimize engagement and diversity in slate recommenders?

While our initial goal encompassed three objectives, we focus on optimizing engagement and diversity in this work. This decision was made for two reasons: first, effectively balancing as little as two objectives in slate recommendation presents significant challenges, and second, these two objectives represent a fundamental tension in recommender systems - between maximizing immediate user engagement and maintaining recommendation diversity. Our findings with two objectives provide important insights that can inform future work on three or more objectives. In order to answer the main research questions, we answer the following questions in our experiments:

RQ1 How effectively does a multi-objective reinforcement learning approach balance multiple objectives (engagement and diversity) compared to single-objective methods?

[1] Created by the author.

RQ2 How scalable is this approach for larger item catalogs and different slate sizes?

In order to optimize multiple objectives simultaneously, we use an existing method introduced by Xu et al. [41] to simultaneously optimize weights using Prediction-Guided Multi-Objective Reinforcement Learning (PGMORL). This method efficiently finds Pareto-optimal policies through a prediction-guided evolutionary strategy, making it particularly suitable for continuous action spaces, which is what we are dealing with. We conduct experiments using a simulator based on synthetic and semi-synthetic embeddings (further explained in 3.3) to demonstrate the usefulness of the introduced model on handling multiple objectives in a dynamic and interactive environment.

2 Related Work

This section provides an overview of related work in the field of slate recommendation, Multi-Objective RL (MORL), and the evaluation methods in these works.

2.1 Slate recommendation

The field of slate recommendation has evolved from early theoretical frameworks to modern scalable approaches. Milani Fard and Pineau [28] first formalized the concept of recommending multiple items simultaneously in an MDP, allowing users to make the final selection and guaranteeing near-optimal policies. While groundbreaking, their approach faced scalability challenges in large state spaces, which is a fundamental problem of slate recommendation [1]. For example, with a slate size of 8 and an item catalog containing 1000 items, the number of possible slates would be approximately 9.7×10^{23}.

Subsequent research focused on addressing these scalability limitations. Sunehag et al. [39] introduced applied deep RL techniques to handle larger action spaces. Their approach successfully handled high-dimensional state and action spaces, but the computational cost of repeatedly evaluating the value function for each slate position made it challenging for real-time industry applications. Ie et al. [20] later introduced a more scalable solution through slate decomposition.

A recent breakthrough came from Deffayet et al. [10] who proposed GeMS, representing slates in a continuous latent space using variational auto-encoders. This approach offers superior scalability and generalization compared to previous methods, though it requires logged interaction data.

This evolution of approaches reflects the field's progress in addressing the fundamental challenge of slate recommendation: maintaining recommendation quality while scaling to large action spaces. Each advancement has traded off different constraints, from computational feasibility to modeling assumptions, in pursuit of practical, effective slate recommendation systems.

2.2 Multi-Objective Reinforcement Learning in Recommendation

While MORL has shown promise in various domains, its application to slate recommendation remains limited. The only previous work in this space by Keat et al. [23] employed a single-policy approach with predetermined linear scalarization. Their method showed improvements in balancing engagement and diversity but lacked flexibility in adapting to changing user preferences. According to the guidelines laid out by Hayes et al. [17], a multi-policy approach is more appropriate when the utility function is not fixed or known in advance. Applying this insight to recommender systems suggests that a multi-policy method, which explores different trade-offs between objectives, provides more flexibility and robustness in adapting to changing user needs.

Prediction Guided MORL (PGMORL) [41] represents a significant advancement in multi-objective reinforcement learning, offering dynamic trade-off adaptation through a multi-policy approach. However, it has not yet been applied to the slate recommendation problem.

2.3 Research Gap and our Approach

Our analysis of existing work reveals several key opportunities in the field. While GeMS offers an efficient solution for slate representation and generation, and PGMORL provides sophisticated multi-objective optimization, no existing work has successfully combined these approaches for slate recommendation. We've identified a significant integration opportunity wherein GeMS's latent representation can be effectively leveraged within PGMORL's framework, while multi-objective optimization can enhance slate recommendation, ultimately offering potential for more flexible and adaptable recommendations. PGMORL's ability to handle continuous action spaces makes it particularly suitable for integration with GeMS's latent slate representations. Our work aims to bridge this gap through two primary contributions: first, by integrating GeMS's slate representation with PGMORL's multi-objective optimization, and second, by evaluating the effectiveness of this combination for slate recommendation while analyzing the trade-offs between different objectives in this combined approach. Our work addresses this gap by implementing and evaluating this combination, providing empirical insights into how modern multi-objective optimization techniques perform in slate recommendation scenarios. This implementation allows us to study important practical questions about the trade-offs between engagement and diversity in slate recommendation, and how effectively these can be balanced using state-of-the-art methods.

3 Background

This chapter introduces the fundamental concepts needed to understand multi-objective slate recommendation: the mathematical framework of Multi-Objective Markov Decision Processes (MOMDP), principles of multi-objective optimization, and approaches to RL evaluation. These concepts form the theoretical foundation for our proposed approach to optimizing multiple objectives in slate recommendation systems.

3.1 Notations and Problem Formulation

We define a slate recommendation scenario where a user interacts with a recommender system (RS). The goal of this system is to optimize long-term engagement while also retaining diversity and novelty in the recommendations. We can optimize these goals by defining them as a separate objective for our system to optimize for. We define a Multi Objective Markov Decision Process (MOMDP) as a tuple $\langle \mathcal{S}, \mathcal{A}, P, \mathbf{R}, \gamma \rangle$, where:

- state $s \in \mathcal{S}$ represents the user state and summarizes information about the past click behavior.
- action $a_t \in \mathcal{A}$ corresponds to a tuple (slate) containing the items $(i_t^1, \ldots, i_t^k)$ where $(i_t^j)_{1 \leq j \leq k}$ are items from the collection $\mathcal{I}$ and k is the size of the slate. Each tuple a_t has a corresponding click tuple $c_t = (c_t^1, \ldots, c_t^k), c_t^j \in \{0, 1\}$, which signifies the click behavior of a user at time step t. The size of all possible slates is of a combinatorial nature: $|\mathcal{A}| = \frac{|\mathcal{I}|!}{(|\mathcal{I}|-k)!}$.
- Transition probabilities $P : \mathcal{S} \times A \times \mathcal{S} \to [0, 1]$
- The reward function $\mathbf{R} : S \times A \to \mathcal{R}^n$ describes a vector of n rewards, one for each objective. This is in contrast to a regular MDP, where the reward function usually outputs a scalar. In this scenario, there are in total 3 objectives: engagement, diversity, and novelty. At state s_{t-1}, a reward vector $\mathbf{r}_t = [r_t^{engagement}, r_t^{diversity}]$ is returned.

Consequently, the value function $\mathbf{V}^\pi \in \mathcal{R}^n$ specifies the expected cumulative discounted reward vector following policy π:

$$\mathbf{V}^\pi = \mathbb{E}\left[\sum_{k=0}^{\tau} \gamma^k \mathbf{r}_{t+k+1} \mid \pi, s_t = s\right] \tag{1}$$

$\mathbf{V}^\pi$ defines the long-term value accumulated over T steps for engagement, diversity, and novelty. The values in this vector represent the long-term value for the respective objective. A high value for the objectives novelty and diversity in $\mathbf{V}^\pi$ indicates that policy π generates diverse and novel recommendations.

Optimizing for $\mathbf{V}^\pi$ is not possible in the same way as for a single objective value function, as there is no unique optimal value $\mathbf{V}^\pi$ [34, 17]. In fact, an increase in item diversity could lead to a decrease in engagement [23].

In order to optimize these objectives simultaneously, we use a linear scalarization function f that projects the multi-objective value $\mathbf{V}^\pi$ to a scalar value:

$$V_{\mathbf{w}}^\pi = f(\mathbf{V}^\pi(s), \mathbf{w}) = \mathbf{w}^T \mathbf{V}^\pi(s) \tag{2}$$

$\mathbf{w}$ signifying a set of weights for each objective. This scalar value can be optimized for different weights for the respective objectives $\mathbf{w}$. Unlike in single objective optimization, a solution set is found where all objectives are optimized without being detrimental to the performance on other objective, i.e. Pareto optimal [17]. This set containing such policies is called a coverage set (CS):

$$CS(\Pi) \subseteq U(\Pi) \wedge (\forall u, \exists \pi \in CS(\Pi), \forall \pi' \in \Pi : u(\mathbf{V})^{\pi} \geq u(\mathbf{V}^{\pi'})) \tag{3}$$

The coverage set is an approximation of what the pareto front would look like. In order to evaluate the (CS), the hypervolume metric is used to determine to compare the increase in accuracy of a Pareto-efficient solution compared to that of a non-dominated reference $\mathbf{V}^{\pi}_{ref}$ [17]:

$$\text{HyperVolume}(CS, \mathbf{V}_{\text{ref}}) = \bigcup_{\pi \in CS} \text{Volume}(\mathbf{V}_{\text{ref}}, \mathbf{V}^{\pi}). \tag{4}$$

Figure 2 Shows the example of how a multi-objective algorithm optimizes for multiple policies.

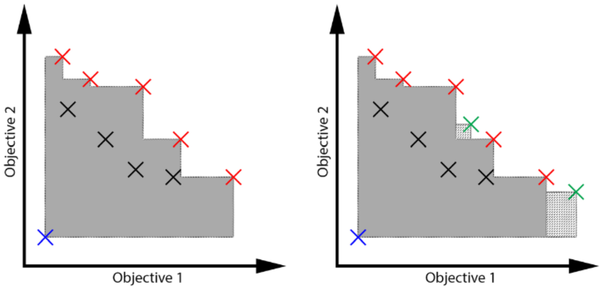


Fig. 2: Visualization of the the coverage set in a two-objective space. The black points represent the dominated set, illustrating solutions that are outperformed by others. The red points show the coverage set, which is a subset of the Pareto Front, capturing essential non-dominated policies necessary for optimal trade-offs based on user preferences. The green points indicate new policies being added to the coverage set, which results in an increased hypervolume, showcasing an improvement in the solution space. Image adapted from Hayes et al. [17]

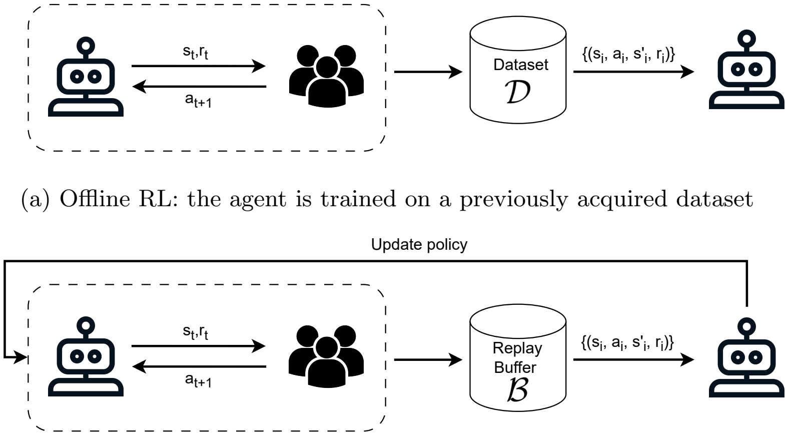
3.2 Diversity and Novelty objective definitions

Diversity [25] note that diversity should be considered during the recommendation process, instead of being applied after the recommendation process. There have been many different ways of defining diversity in recommender systems. The most widely used method is intra-list diversity (ILD) [43, 25], which measures the pairwise item diversity within a slate using a distance metrics such as cosine distance, Jaccard distance or euclidean distance.

Cui et al. [9] formulated diversity using Shannon's entropy, in order to define topic distributions inside a recommendation list. Shannon's entropy measures the uncertainty or randomness in a set of outcomes, which can be interpreted as diversity in the context of topic distributions. Keat et al. [23] defined diversity in their MORL-algorithm using a variant based on Shannon's entropy.

Novelty Novelty has been defined as the inverse of popularity, signifying items with few interactions [4]. Keat et al. [23] defined novelty in their work as an objective using this definition. Ge et al. [15] Stated in their work that catalog coverage can be defined by using the catalog coverage. Catalog coverage is a measure that can be particularly helpful for systems to evaluate whether a system over time recommends enough novel items. With small catalog sizes (¡1000) it is a good measure for whether the recommender system recommends enough novel items over time [18].

3.3 Evaluating RL models



(a) Offline RL: the agent is trained on a previously acquired dataset

(b) Online RL: While the agent is training, new data is acquired and stored in the replay buffer

Fig. 3: The difference between online and offline RL.

RL-based systems can be trained using an online or offline approach. The online approach, i.e. testing the agent in a real-life setting, is the most empirically indicative method on whether a method works or not, but this approach has many caveats. It is expensive, time consuming and training a policy in an online environment hurts the user experience [26, 8]. Offline RL, as shown in figure

3, involves training an RL agent on a historical dataset, and thus avoids these practical issues. This however introduces its own challenges. This approach can result in value overestimation, is susceptible to biases in logged data, and can lead to myopic evaluation [14, 11]. Using a simulator for evaluation offers numerous benefits:

- Supplementing real-world data in the recommender system (RS) training and testing process with synthetic analogues in a simulated environment.
- Preserving the privacy of real-world data [38],
- Enabling counterfactual analysis.
- Counteracting biases in datasets [6].
- Allowing the testing of a new RS under different conditions and within various scenarios of user behavior [19].

While synthetic item embeddings are more convenient to acquire, they provide a less realistic representation of item catalogs. Semi-synthetic embeddings partially address this by combining real item embeddings from existing datasets with generated user embeddings based on assumed genre preferences. Though this still represents a simplified version of real-world complexity, it provides a controlled environment for systematic evaluation.

4 Methodology

This chapter describes our approach to multi-objective slate recommendation and how we evaluate it. We first present our technical solution that combines methods in the literature to optimize multi-objective slate recommendation. We then detail our implementation choices and experimental design for evaluating the effectiveness of this approach.

4.1 Base Algorithms

Soft Actor-Critic (SAC) is an off-policy actor-critic algorithm that optimizes expected reward while maximizing entropy [16]. Its off-policy learning provides sample efficiency and entropy maximization maintains stability, making it particularly suitable for continuous action spaces. SAC can be combined with GeMS, allowing SAC to operate in the slate encoded latent space.

Prediction Guided MORL (PGMORL), introduced by Xu et al. [41], maintains a population of policies and uses a prediction-guided evolutionary strategy to find Pareto-optimal policies. In each generation, it fits analytical models to predict potential improvements and selects the most promising policy-weight combinations to optimize. The original PGMORL implementation used PPO as its base RL algorithm. For our slate recommendation problem, we replace PPO with SAC+GeMS while keeping PGMORL's evolutionary and prediction mechanisms,as SAC provides better sample efficiency through off-policy learning and effectively handles continuous action spaces.

4.2 Multi-Objective Slate Optimization

Our approach combines PGMORL with SAC+GeMS to optimize both engagement, diversity, and novelty objectives in slate recommendation. We use PGMORL's framework to maintain a population of policies, each optimizing for different trade-offs between these objectives. While the original PGMORL used PPO, we adopt SAC as our base RL algorithm for its sample efficiency in continuous action spaces. In each generation of training, PGMORL's prediction model identifies promising policy-weight combinations that could improve our current Pareto front. Selected policies are then optimized using SAC with their corresponding objective weights, where SAC generates a continuous action which is converted to a slate using TopK selection. This combination leverages PGMORL's ability to find diverse Pareto-optimal policies while using SAC+GeMS efficient slate construction mechanism. The result is a set of policies that offer different trade-offs between engagement, diversity, and novelty objectives.

To evaluate our approach in a controlled yet realistic environment, we extend the SARDINE simulator [12] to support multi-objective optimization. While SARDINE was originally designed for single-objective slate recommendation, we adapt it to provide reward signals for both engagement and diversity objectives, allowing us to test the long-term effects of our multi-objective optimization approach.

4.3 Preliminary Analysis and Method Selection

Our initial approach aimed to combine GeMS's latent slate representation with PGMORL for multi-objective optimization. However, despite extensive pretraining efforts, our implementation of SAC+GeMS showed unexpectedly low engagement metrics. While GeMS demonstrated promising results in the original paper by Deffayet et al. [10], we were unable to replicate this success in our setup, despite doing an extensive parameter grid search on hyperparameters β and γ. Given these challenges, we pivoted to using SAC+TopK as our foundation, which showed more reliable performance during training. This led to our final approach: integrating PGMORL with SAC+TopK to create a multi-objective slate recommendation system. SAC+TopK, also introduced by Deffayet et al. [10], takes a simpler approach where SAC outputs a continuous action in the item embedding space, and the k items closest to this embedding (according to dot product) are selected for the slate. This provides a computationally efficient way to handle slate construction and is used instead used to form the proposed multi-objective solution.

4.4 Reward signals

Engagement corresponds to the amount of interaction that a user has with a system. This can be defined as dwell time, click rate and session length [1, 45]. In this scenario, we define the reward signal for the engagement objective simply as the amount of clicks within a recommended slate, following convention [10]:

$$r^{clicks} = \sum_{j=1}^{k} c_t^j, \tag{5}$$

We define the reward signal for diversity using ILD, with cosine distance as our distance metric. The reward signal for diversity is defined using a distance based similarity approach, as introduced by Ziegler et al. [44], commonly called Intra-List Diversity, signifying the average distance between two items within a list. This approach is one of the most used ways to define diversity metrics [40, 25, 4]:

$$r_t^{div} = \frac{1}{|a_t|(|a_t|-1)} \sum_{i \in a_t} \sum_{j \in a_t} d(i,j) \tag{6}$$

Where d is a distance measure. We will use the cosine distance. This measure is frequently used for items represented as topic vectors, and it is sensible to do so, since items are rated on a continuous scale from 0 to 1 whether to ascribe topic proclivity. This cosine distance has also been used by Anderson et al. [2] for calculating the distance between topic vectors.

5 Experimental design

In this section, we detail the experimental setup designed to comprehensively evaluate our proposed multi-objective approach for multi-objective slate recommendation. The experiments are structured to examine how well PGMORL can balance engagement and diversity as simultaneous objectives (RQ1) when compared to methods that focus on single objectives. Additionally, we investigate the scalability of this approach in the realm of CPU and GPU memory usage, and runtime (RQ2), particularly its effectiveness when applied to larger item catalogs and various slate sizes.

5.1 Environment

As explained by the reasons noted in section 3.3, We use the SARDINE simulator for our experiments (figure 4), which provides a controlled recommendation environment with dynamic user behavior. Semi-synthetic datasets are used, combining item embeddings from the MovieLens-100k (ML-100k) dataset (100,000 ratings from 1000 users on 1682 movies) [29] with generated user embeddings (more information on the MovieLens-100k dataset can be found in Appendix A.4). Item embeddings can be directly obtained from item-topic assignments from the datasets. The simulator implements a click model based on relevance scoring which includes a boredom mechanism to penalize short-term recommendation strategies (see Appendix A.2 for implementation details). User behavior is modeled through topic-based preferences and click probabilities (Appendix

A.1), creating a dynamic environment suitable for evaluating long-term recommendation strategies. The generation of user embeddings and specific parameter settings are detailed in Appendix A.3.

The catalog size is within the magnitude of current RL-based slate recommender systems (≈ 1000). The simulator replicates a recommendation platform where users are recommended a slate of movies based on their historical preferences.

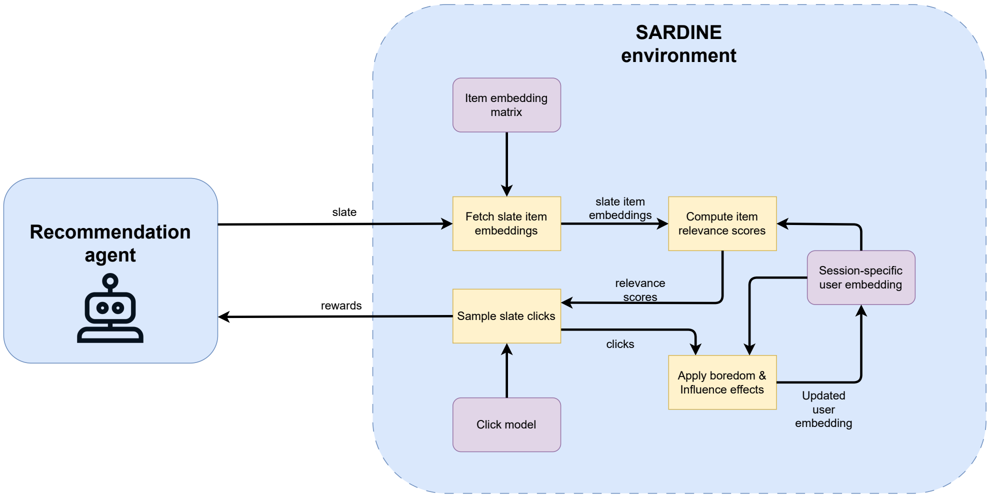


Fig. 4: Diagram summarizing the different components of the SARDINE simulator.

Using this simulator and basing our environment on it using synthetic and semi-synthetic datasets will contribute to our goal to test the feasibility of balancing multiple objectives in a dynamic recommendation setting.

5.2 Baselines

To assess the performance of the MORL agent, we compare it against state-of-the-art baselines that are available in the field of RL-based recommender systems. As no other MORL technique has been proposed in earlier slate RLRS work, we can only compare it to Single objective algorithms:

1. **Single-Objective SAC**: A Soft-Actor Critic (SAC) agent optimizing solely for one objective, representing a standard single-objective RL approach. An off-policy actor-critic deep RL algorithm that optimizes the expected reward while also maximizing entropy [16]. To compare this baseline with the

Multi-Objective method, two variants with different reward signals are included: one applying the diversity reward signal (ILD) and the other with engagement reward signal (clicks).

2. **REINFORCE**: This algorithm employs a simple yet effective policy gradient method, which has been scaled to handle millions of items in the implementation of Chen et al. [7]. The model estimates the value of individual items rather than taking into account the full slate, in a similar vein to [20].
3. **Random**: A naive baseline that generates slates randomly, serving as a lower bound on performance.
4. **Short-Term Greedy Oracle**: This baseline maximizes immediate reward by picking the optimal slate maximizing the relevance function defined in section 5.1. In a non-dynamic setting (i.e. user shows no dynamic behavior like boredom), this model serves as a theoretical max.

5.3 Scalability Tests

In order to evaluate the formation of the pareto front of the MORL algorithm (RQ1) and scalability of this approach (RQ2), we vary both the catalog size $|\mathcal{I}|$ and slate size k:

- Catalog sizes $\mathcal{I}$: 100, 500, 1000, 1682 (ml-100k dataset) items. We use both purely synthetic data and semi-synthetic data based on the MovieLens dataset to cover a range of catalog sizes.
- Slate sizes k: 3, 10, 20 recommended items. These slate sizes cover typical real-world scenarios, from small carousels to full pages of recommendations.

5.4 Training protocol

Each algorithm is trained for 500,000 steps, witch one episode representing a user traversing $\tau = 100$ recommendation moments (steps). The reason for this set number is because previous research [10] conducted showed that this is an adequate amount of training steps for most baselines to achieve a good return on objective, and we furthermore want to test which model is most sample efficient. Every 5,000 training steps, the agents are tested on a validation set of 25 user trajectories. To assess statistical significance, we run each experimental baseline with 5 different random seeds. In the case of single-objective agents, the model checkpoint giving the highest return on the validation episodes is tested on 500 test-user trajectories. In the case of PGMORL, the resulting non-dominating policies are tested on 500 test-user trajectories as well.

Experiments are run using the DAS-6 cluster [3]. Each experiment was on a dedicated machine with an A4000 GPU while not peforming any other tasks. The amount of used GPU memory and CPU memory are logged in consistent intervals during training.

To answer our research question RQ1, we compare the Pareto front of our proposed MORL approach versus the baselines to assess the effectiveness of multi-objective optimization. For RQ2, we compare the evaluation metrics for this RQ over slate size 3, 10, and 20.

5.5 Evaluation metrics

RQ1 Engagement is evaluated on the long-term value generated on the click reward signal, namely the cumulative clicks:

$$\text{Long-term Engagement} = \sum_{k=1} r_{k+1}^{\text{clicks}} \tag{7}$$

This corresponds to the value function for the engagement objective with $\gamma = 1$.

Diversity is evaluated by determining the average intra-list diversity over all evaluated users.

$$\text{Average ILD} = \frac{1}{|U|} \cdot \sum_{u \in U} \frac{1}{\tau_u} \sum_{t \in \tau_u} r_t^{\text{div}} \tag{8}$$

With U signifying the array of evaluated users.

Novelty is defined as catalog coverage and reflects the degree to which the generated recommendations cover the catalog of available items [15]. The catalog coverage, defined as the percentage of available items which are recommended to a user is defined as follows:

$$\text{Catalog coverage} = \frac{|\bigcup_{t=1 \ldots N} \mathcal{I}_{a_t}|}{|\mathcal{I}|} \tag{9}$$

Even though the return for this objective is not optimized using our MORL solution, we will log data so that we can investigate the effects of optimizing diversity and engagement on this objective.

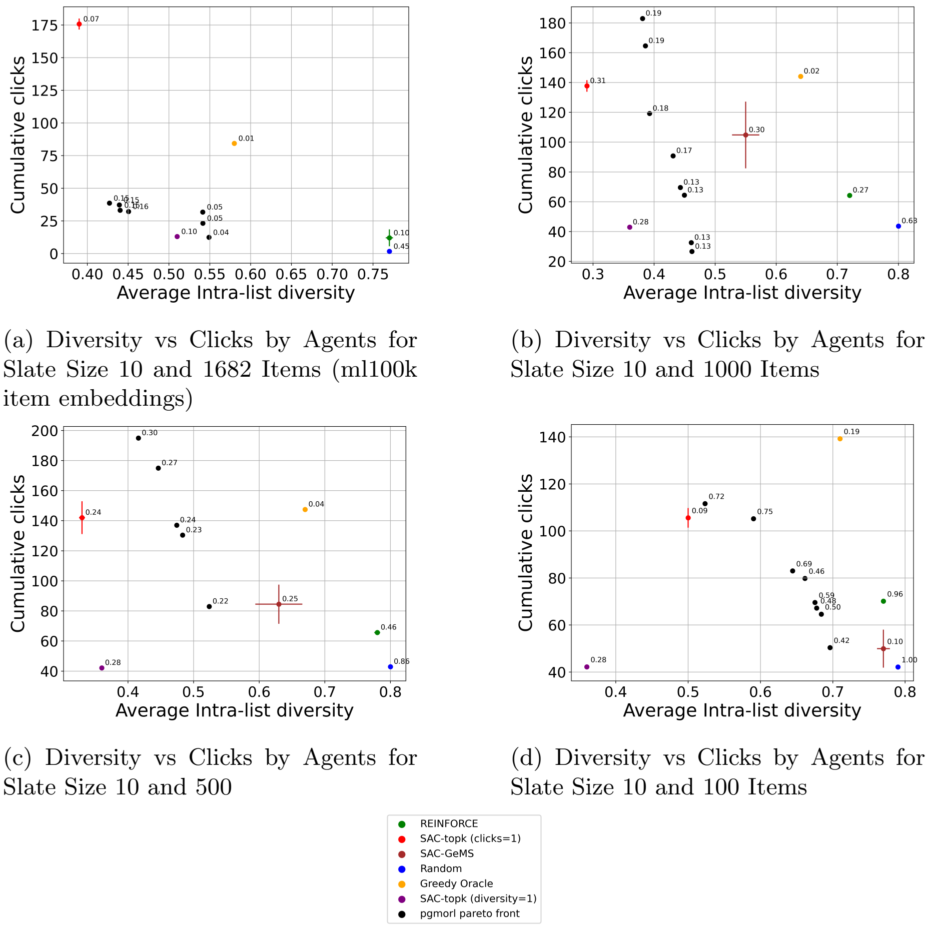
RQ2 During training, the GPU and CPU memory usage will be logged, defined in Gigabytes (GB). Furthermore, runtime will be tracked in hours (h).

6 Results

Our experiments evaluate the effectiveness of PGMORL in optimizing multiple objectives for slate recommendation (RQ1) and assess its scalability across different catalog and slate sizes (RQ2). We present our findings organized by our key research questions.

6.1 Multi-Objective Optimization Performance (RQ1)

Figure 5 illustrates the performance trade-offs between diversity and engagement (clicks), across different catalog sizes. Each point represents a different policy, with numbers indicating the catalog coverage percentage achieved. More results can be found in Appendix B.



(a) Diversity vs Clicks by Agents for Slate Size 10 and 1682 Items (ml100k item embeddings)

(b) Diversity vs Clicks by Agents for Slate Size 10 and 1000 Items

(c) Diversity vs Clicks by Agents for Slate Size 10 and 500

(d) Diversity vs Clicks by Agents for Slate Size 10 and 100 Items

Fig. 5: Charts illustrating the performance of the agents over multiple slate size and item catalogs. The numbers next to the agent signify the catalog coverage

Pareto Front Analysis (RQ1) PGMORL consistently produces a clear Pareto front across all catalog sizes (100-1682 items), demonstrating its ability to find diverse trade-offs between objectives. The Pareto front shows that:

- Higher click rates generally come at the cost of reduced diversity, confirming the inherent tension between these objectives
- PGMORL discovers policies spanning from high-diversity/low-clicks to high-clicks/low-diversity, offering flexible deployment options
- The single-objective baselines (SAC-GeMS and SAC-top K) typically appear as individual points near the extremes of the Pareto front

When compared to the baselines, we found that a random policy achieves the highest diversity while the Greedy Oracle achieves the highest click rate, at the cost of diversity. These results are in line with our expectations.

- Random policy achieves high diversity but poor engagement across all configurations
- Greedy Oracle achieves high click rates but at the cost of diversity
- REINFORCE performs similarly to SAC variants but with slightly lower performance in both objectives
- PGMORL consistently finds policies that dominate single-objective approaches in terms of Pareto efficiency for small catalog sizes.

6.2 Scalability Analysis (RQ2)

Table 1 presents the computational cost across different approaches:

Agent	Training time (h)	Peak GPU Memory Usage (GB)	Average GPU Memory Usage (GB)	Peak CPU Memory Usage (GB)	Average CPU Memory Usage (GB)
PGMORL	2.53	60.72	47.66	27.79	18.71
SAC-GeMS	1.17	7.66	3.68	6.22	5.69
SAC-top K	1.66	6.62	6.61	5.88	5.67
REINFORCE	1.11	10.53	10.52	5.40	5.16
HAC	1.8	9.09	8.79	5.49	5.36
Random	0.03	0.00	0.00	0.02	0.08
Greedy Oracle	0.03	0.00	0.00	0.02	0.08

Table 1: Average training time and other statistics on performance, run on the synthetic item embeddings for slate size = 10, number of items = 100, timesteps=500,000

Training Efficiency PGMORL requires significantly more training time (2.53 hours) compared to single-objective approaches (1.11-1.66 hours), while also needing more memory (60.72 GB memory at peak). Single-objective methods maintain relatively modes memory requirements (6-10 GB GPU memory at peak). CPU memory usage follows a similar pattern, with PGMORL requiring 3-5x more memory. CPU memory usage follows a similar pattern, with PGMORL requiring 3-5x more memory.

Catalog Size Impact (RQ2) The plots in Figure 5 demonstrate how performance scales across different catalog sizes: Performance patterns remain consistent across catalog sizes, suggesting good scalability of the core algorithm. Larger catalogs (1000-1682 items) show a wider range of trade-offs represented by the solutions in comparison to smaller catalogs, while Smaller catalogs (100 items) exhibit tighter clustering of solutions. This implies that the algorithm explores a wider variety of optimal trade-offs as the catalog size increases.

6.3 Coverage Analysis

The catalog coverage results (annotated numbers in Figure 5) reveal that PGMORL policies achieve varying coverage levels (30-70%) depending on their position on the Pareto front. Higher diversity policies generally correlate with higher catalog coverage. Single-objective approaches tend to have lower coverage, particularly when optimizing for clicks alone. The random baseline achieves high coverage but at the cost of other objectives, which is to be expected for a random baseline. The greedy oracle recommends the items that have in the highest click probability by definition, so the catalog coverage is naturally low.

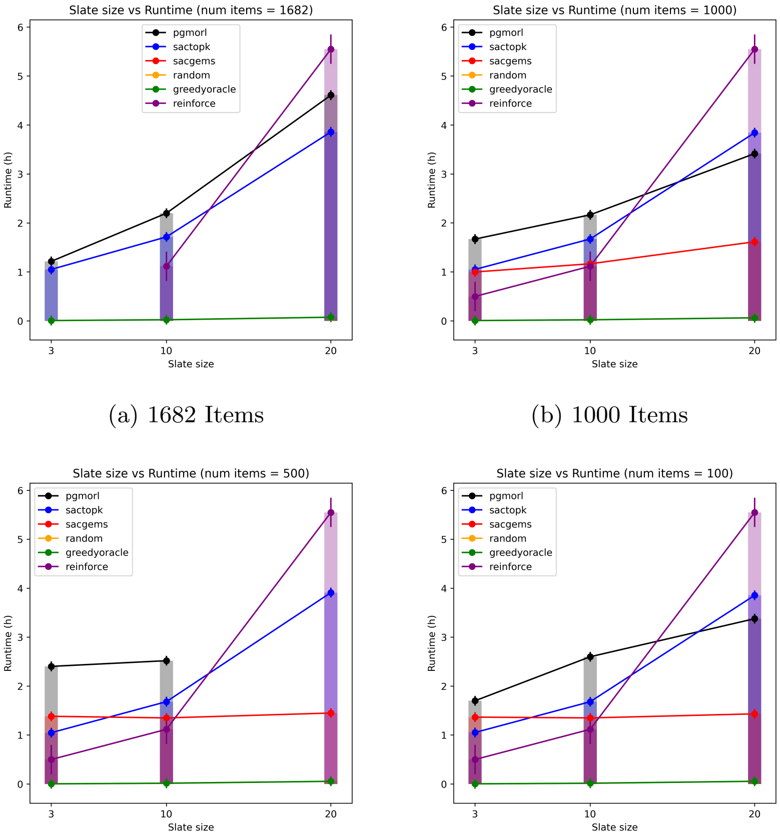
6.4 Runtime Analysis Across Slate Sizes

Figure 6 shows the runtime performance across different slate sizes (3, 10, and 20) and catalog sizes (1000, 500, and 100 items). The runtime measurements reveal several key patterns:

Impact of Slate Size All methods show increased runtime as slate size increases, but with varying degrees of scaling. REINFORCE exhibits the steepest increase in runtime, particularly for slate sizes greater than 10. PGMORL shows moderate scaling, with runtime increasing roughly linearly with slate size. PGMORL maintains reasonable runtime, similar to SAC-topk growth despite handling multiple objectives This is to be expected, as SAC-topk is implemented in PGMORl. SAC-based methods (SAC-top-k and SAC-GeMS). SAC-GeMS demonstrates the best scaling among learning-based methods, particularly for larger slate sizes. The random and greedy oracle baselines maintain constant low runtime regardless of slate size.

Catalog Size Effects Larger catalogs (1000 items, Fig. 6b) show more pronounced runtime differences between methods. For medium-sized catalogs (500 items, Fig. 6b), the runtime gap between methods narrows.With small catalogs (100 items, Fig. 6c), the relative performance differences persist but with reduced absolute differences.

Method-Specific Performance The results indicate that while PGMORL requires more computational resources than single-objective methods (as shown in Table 1), its runtime scaling with slate size remains manageable. This suggests that PGMORL provides a practical solution for multi-objective optimization in slate recommendation, even as the problem size increases. However, for very large slate sizes (20+), additional optimization techniques may be necessary to maintain reasonable computation times.



(a) 1682 Items

(b) 1000 Items

(c) 500 Items

(d) 100 Items

Fig. 6: Runtime difference per slate size for all baselines

7 Conclusion

This work set out to investigate whether multi-objective reinforcement learning can effectively balance multiple objectives simultaneously (RQ1) while also investigating its computational feasibility over a range of item catalog sizes and slate sizes (RQ2).

while we have carefully constructed our methodology and experimental setup to ensure these goals, several important limitations and considerations must be addressed.

7.1 Limitations

A number of important limitations that may threaten the validity of the findings in the experiments should be acknowledged. A significant limitation of this study is the use of ideal item embeddings in the experiments by the models. We cannot assume the availability of such embeddings in a more realistic recommender scenario. During testing, when the ideal item embeddings were replaced by item embeddings learned by Matrix factorization [24], the recommendation performance for all models was drastically reduced. Worse performance was to be expected as is shown in previous work [10]. However, the models using MF-embeddings were not performing better than the random baseline on all slate size - catalog size configurations, which was worse than expected. In future work, the sub-optimal results could be solved by training these embeddings on a larger dataset, as 100,000 user trajectories is not sufficient for sufficient performance.

Secondly, while the simulator provides a controlled environment for testing, it makes several simplifying assumptions about user behavior. One limitation is that user boredom was modeled only through repeated exposure to the same main topic, which restricts the portrayal of more complex interaction patterns.

The click model used in the experiments did not account for the influence effect: user profiles remained static, except for the boredom effect, missing the dynamic evolution of user interests. The reason for not using this interaction pattern in the experiment is because using it would likely require additional training time for the optimization of the objectives.

Furthermore, the ranking order of items within the recommended slate was not accounted for in the experiment This means that the influence of item positions on user engagement was not taken into account, potentially impacting the realism of user interaction modeling. In practical scenarios, users may exhibit varying behaviors depending on the position of items within a slate, which can affect click probabilities and overall engagement.

Additionally, it may be more challenging to analyze the interactions between ranking and other objectives, such as diversity. a click model that reduces click probability of an item based on its position in the slate is already available in the Sardine simulator that could be used for future research [12]. More training would be expected to be necessary for the models to optimize the objectives.

Moreover, user profiles remained static and did not change with repeated item interactions, missing the dynamic evolution of user interests. Incorporating such an influence effect could improve long-term outcome predictions and better simulate dynamic real-world behavior.

This study assumes that each user trajectory lasts 100 recommendation steps. Using this assumption, it was possible to find a part of the pareto front in the case of PGMORL. In VirtualTaoBao, the user simulation had the possibility to preemptively quit a session [37]. Implementing variable-length sessions could introduce complexity but would enhance realism.

Similarly, distinguishing between different user types was omitted due to added training requirements. In the work of Anderson et al., the difference was noted between ”generalist” and ”specialist” users [2]. This notion could have been an interesting way of implementing different types of users, but this was not done, since this would have required additional training for the models to optimize well. In future research, when evaluating the models, it would be interesting to take into account this notion of differing user types. An RL agent could optimize this by prediction the possible user type which can be added as information to the user state.

Despite the limitations discussed, our findings provide valuable insights into the challenges and potential of multi-objective optimization in slate recommendation. The balanced approach of PGMORL highlights its capability to manage engagement and diversity effectively, setting the stage for further research to address these complexities and refine the methodology for broader real-world applications.

In this work, the assumption was made that the utility function is linear between the objectives. This however might not be true making a trade-off between objectives in the case of diversity and engagement, one can argue that using a linear weighting function might not reflect the actual relationship between these two objectives. If users' preferences are non-linear, a linear utility function is unable to accurately represent these preferences [34].

While this work focused on optimizing engagement and diversity, future work could extend our approach to include novelty as a third objective. Our results suggest that handling multiple objectives in slate recommendation is challenging but feasible, laying the groundwork for incorporating additional objectives.

7.2 Practical Implications

Despite these limitations, our findings have several important implications for real-world recommender systems. The Pareto front of solutions enables system operators to dynamically adjust the trade-off between objectives based on business needs or user preferences. For instance, a streaming service could favor diversity during content discovery phases while prioritizing engagement for personalized recommendations, all using the same underlying model.

This thesis introduced and evaluated a multi-objective reinforcement learning approach, PGMORL, for slate recommendation, aiming to balance engagement and diversity. Our findings offer guidance for implementing multi-objective slate recommendations, where diverse suggestions can enhance content discovery and catalog usage. While PGMORL successfully identifies trade-offs between engagement and diversity, challenges remain in consistently producing well-distributed Pareto fronts, particularly with larger catalogs. Although the approach scales well with slate size, it requires more computational resources than single-objective methods.

References

[1] M. Mehdi Afsar, Trafford Crump, and Behrouz Far. *Reinforcement learning based recommender systems: A survey.* arXiv:2101.06286 [cs]. June 2022. DOI: `10.48550/arXiv.2101.06286`. URL: `http://arxiv.org/abs/2101.06286` (visited on 01/23/2024).

[2] Ashton Anderson et al. "Algorithmic effects on the diversity of consumption on spotify". In: *Proceedings of the web conference 2020.* 2020, pp. 2155–2165.

[3] Henri Bal et al. "A medium-scale distributed system for computer science research: Infrastructure for the long term". In: *Computer* 49.5 (2016), pp. 54–63.

[4] Pablo Castells, Neil Hurley, and Saul Vargas. "Novelty and diversity in recommender systems". In: *Recommender systems handbook.* Springer, 2021, pp. 603–646.

[5] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. "How algorithmic confounding in recommendation systems increases homogeneity and decreases utility". In: *Proceedings of the 12th ACM conference on recommender systems.* 2018, pp. 224–232.

[6] Jiawei Chen et al. "Bias and debias in recommender system: A survey and future directions". In: *ACM Transactions on Information Systems* 41.3 (2023), pp. 1–39.

[7] Minmin Chen et al. "Top-k off-policy correction for a REINFORCE recommender system". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* 2019, pp. 456–464.

[8] Xiaocong Chen et al. *A Survey of Deep Reinforcement Learning in Recommender Systems: A Systematic Review and Future Directions.* arXiv:2109.03540 [cs]. Sept. 2021. DOI: `10.48550/arXiv.2109.03540`. URL: `http://arxiv.org/abs/2109.03540` (visited on 01/23/2024).

[9] Laizhong Cui et al. "A novel multi-objective evolutionary algorithm for recommendation systems". In: *Journal of Parallel and Distributed Computing* 103 (2017), pp. 53–63.

[10] Romain Deffayet et al. "Generative Slate Recommendation with Reinforcement Learning". en. In: *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining.* Singapore Singapore: ACM, Feb. 2023, pp. 580–588. ISBN: 978-1-4503-9407-9. DOI: `10.1145/3539597.3570412`. URL: `https://dl.acm.org/doi/10.1145/3539597.3570412` (visited on 02/29/2024).

[11] Romain Deffayet et al. "Offline evaluation for reinforcement learning-based recommendation: a critical issue and some alternatives". In: *ACM SIGIR Forum.* Vol. 56. 2. ACM New York, NY, USA. 2023, pp. 1–14.

[12] Romain Deffayet et al. "SARDINE: Simulator for Automated Recommendation in Dynamic and Interactive Environments". In: *ACM Trans. Recomm. Syst.* 2.3 (June 2024). DOI: `10.1145/3656481`. URL: `https://doi.org/10.1145/3656481`.

[13] Simen Eide, David S. Leslie, and Arnoldo Frigessi. *Dynamic Slate Recommendation with Gated Recurrent Units and Thompson Sampling.* arXiv:2104.15046 [cs, stat]. Apr. 2021. DOI: `10.48550/arXiv.2104.15046`. URL: `http://arxiv.org/abs/2104.15046` (visited on 02/29/2024).

[14] Chongming Gao et al. "Advances and challenges in conversational recommender systems: A survey". In: *AI Open* 2 (Jan. 2021), pp. 100–126. ISSN: 2666-6510. DOI: `10.1016/j.aiopen.2021.06.002`. URL: `https://www.sciencedirect.com/science/article/pii/S2666651021000164` (visited on 03/01/2024).

[15] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. "Beyond accuracy: evaluating recommender systems by coverage and serendipity". In: *Proceedings of the fourth ACM conference on Recommender systems.* 2010, pp. 257–260.

[16] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning.* PMLR. 2018, pp. 1861–1870.

[17] Conor F Hayes et al. "A practical guide to multi-objective reinforcement learning and planning". In: *Autonomous Agents and Multi-Agent Systems* 36.1 (2022), p. 26.

[18] Jonathan L Herlocker et al. "Evaluating collaborative filtering recommender systems". In: *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), pp. 5–53.

[19] Eugene Ie et al. *RecSim: A Configurable Simulation Platform for Recommender Systems.* arXiv:1909.04847 [cs, stat]. Sept. 2019. DOI: `10.48550/arXiv.1909.04847`. URL: `http://arxiv.org/abs/1909.04847` (visited on 02/15/2024).

[20] Eugene Ie et al. "SlateQ: A Tractable Decomposition for Reinforcement Learning with Recommendation Sets". en. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.* Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 2592–2599. ISBN: 978-0-9992411-4-1. DOI: `10.24963/ijcai.2019/360`. URL: `https://www.ijcai.org/proceedings/2019/360` (visited on 02/29/2024).

[21] Anna-Katharina Jung et al. "Click me...! The influence of clickbait on user engagement in social media and the role of digital nudging". In: *Plos one* 17.6 (2022), e0266743.

[22] Marius Kaminskas and Derek Bridge. "Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7.1 (2016), pp. 1–42.

[23] Ee Yeo Keat et al. "Multiobjective Deep Reinforcement Learning for Recommendation Systems". en. In: *IEEE Access* 10 (2022), pp. 65011–65027. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2022.3181164`. URL: `https://ieeexplore.ieee.org/document/9791369/` (visited on 02/29/2024).

[24] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix factorization techniques for recommender systems”. In: *Computer* 42.8 (2009), pp. 30–37.

[25] Matevž Kunaver and Tomaž Požrl. “Diversity in recommender systems–A survey”. In: *Knowledge-based systems* 123 (2017), pp. 154–162.

[26] Yuanguo Lin et al. “A Survey on Reinforcement Learning for Recommender Systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). arXiv:2109.10665 [cs], pp. 1–21. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2023.3280161. URL: http://arxiv.org/abs/2109.10665 (visited on 01/23/2024).

[27] Leigh McAlister and Edgar Pessemier. “Variety seeking behavior: An interdisciplinary review”. In: *Journal of Consumer research* 9.3 (1982), pp. 311–322.

[28] M. Milani Fard and J. Pineau. “Non-Deterministic Policies in Markovian Decision Processes”. en. In: *Journal of Artificial Intelligence Research* 40 (Jan. 2011), pp. 1–24. ISSN: 1076-9757. DOI: 10.1613/jair.3175. URL: https://jair.org/index.php/jair/article/view/10682 (visited on 02/26/2024).

[29] *MovieLens*. en. Sept. 2013. URL: https://www.grouplens.org/datasets/movielens/ (visited on 02/29/2024).

[30] Tien T Nguyen et al. “Exploring the filter bubble: the effect of using recommender systems on content diversity”. In: *Proceedings of the 23rd international conference on World wide web*. 2014, pp. 677–686.

[31] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[32] Puthankurissi S Raju. “Optimum stimulation level: Its relationship to personality, demographics, and exploratory behavior”. In: *Journal of consumer research* 7.3 (1980), pp. 272–282.

[33] Yi Ren et al. “Slate-Aware Ranking for Recommendation”. In: *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 2023, pp. 499–507.

[34] Diederik M Roijers et al. “A survey of multi-objective sequential decision-making”. In: *Journal of Artificial Intelligence Research* 48 (2013), pp. 67–113.

[35] Wilbert Samuel Rossi, Jan Willem Polderman, and Paolo Frasca. “The closed loop between opinion formation and personalized recommendations”. In: *IEEE Transactions on Control of Network Systems* 9.3 (2021), pp. 1092–1103.

[36] Guy Shani et al. “An MDP-based recommender system.” In: *Journal of machine Learning research* 6.9 (2005).

[37] Jing-Cheng Shi et al. “Virtual-Taobao: Virtualizing Real-World Online Retail Environment for Reinforcement Learning”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019). Number: 01, pp. 4902–4909. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.33014902.

URL: https://ojs.aaai.org/index.php/AAAI/article/view/4419 (visited on 02/29/2024).

[38] Manel Slokom. "Comparing recommender systems using synthetic data". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. 2018, pp. 548–552.

[39] Peter Sunehag et al. *Deep Reinforcement Learning with Attention for Slate Markov Decision Processes with High-Dimensional States and Actions*. arXiv:1512.01124 [cs]. Dec. 2015. URL: http://arxiv.org/abs/1512.01124 (visited on 02/28/2024).

[40] Haolun Wu et al. "Result Diversification in Search and Recommendation: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[41] Jie Xu et al. "Prediction-guided multi-objective reinforcement learning for continuous robot control". In: *International conference on machine learning*. PMLR. 2020, pp. 10607–10616.

[42] Xing Zhao, Ziwei Zhu, and James Caverlee. "Rabbit holes and taste distortion: Distribution-aware recommendation with evolving interests". In: *Proceedings of the Web Conference 2021*. 2021, pp. 888–899.

[43] Yuying Zhao et al. "Fairness and Diversity in Recommender Systems: A Survey". In: *ACM Trans. Intell. Syst. Technol.* (May 2024). Just Accepted. ISSN: 2157-6904. DOI: 10.1145/3664928. URL: https://doi.org/10.1145/3664928.

[44] Cai-Nicolas Ziegler et al. "Improving recommendation lists through topic diversification". In: *Proceedings of the 14th international conference on World Wide Web*. 2005, pp. 22–32.

[45] Lixin Zou et al. "Reinforcement Learning to Optimize Long-term User Engagement in Recommender Systems". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 2810–2818. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330668. URL: https://dl.acm.org/doi/10.1145/3292500.3330668.

A Simulator technical details

A.1 Click model Implementation

In the SARDINE simulator, a click model is used to represent the behavior of a user. The relevance of items that are presented to a user based on their interests is calculated by applying the dot-product to the user embedding and item embedding:

$$\text{topic relevance} = rel(i,u) = \boldsymbol{e}_i^T \boldsymbol{e}_u \tag{10}$$

Through this relevance score, the item attractiveness of item i for user u is calculated.

$$A_{u,i} = \alpha \cdot \sigma(\text{rel}(i,u)) \quad \text{where} \quad \sigma(x) = \frac{1}{1+\exp^{-\lambda(x-\mu)}}. \tag{11}$$

The attractiveness function is defined as a sigmoid function times α. hyperparameter α adjusts the range of the attractiveness score. In the sigmoid function, λ controls the steepness of the sigmoid curve, i.e., the larger the value, the steeper the sigmoid will be. μ is a shift parameter that ensures the outcome of the sigmoid will be close to 1 if an item is highly matching to a user, and 0 if it is not fitting for a user.

A.2 Boredom mechanism

In order to create a dynamic environment in which greedy algorithms do not result in an optimal outcome. It is a reasonable assumption that a user's interest in said item decreases, as shown by Anderson et al. [2]. Deffayet et al. [12] defined temporary-loss-of-interest boredom in their simulator. The user u who is bored with respect to topic T has their user embedding component $e_{u,T}$ set to 0 for t_b time steps. In this environment, t_b is set to 5.

A.3 User embedding generation based on the ML-100k dataset

As described in the paper introducing SARDINE, [12], Topic interests for a user u are sampled from a categorical non-uniform prior p_t. the probability for a topic j is defined as the ratio of the average number of likes for items with category j divided by the average number of likes with any category.

$$p_{\mathcal{T}}(j) = \frac{\frac{1}{|I_j|}\sum_{i\in I_j} \#\text{likes}(i)}{\sum_{j'\in\mathcal{T}} \frac{1}{|I_{j'}|}\sum_{i\in I_{j'}} \#\text{likes}(i)} \tag{12}$$

A.4 Statistical information about MovieLens-100k dataset

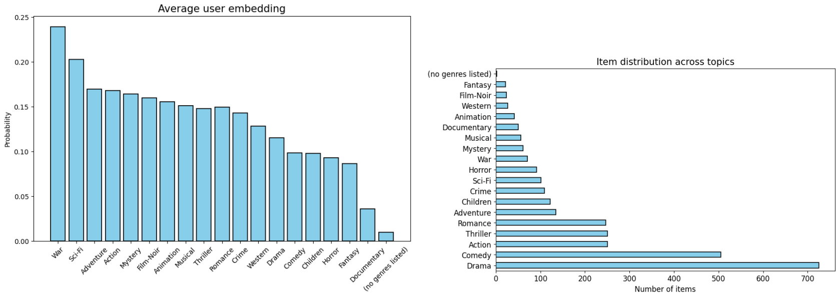
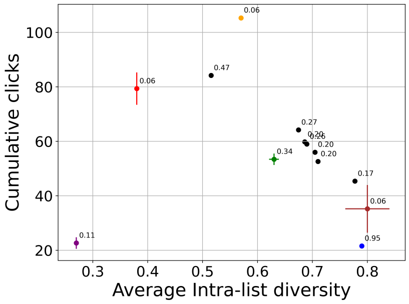
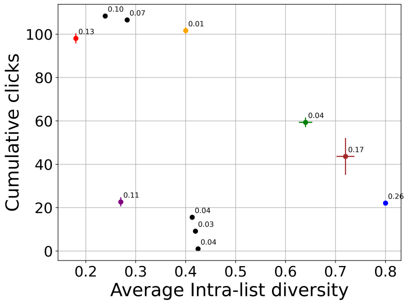


Fig. 7: Charts showing the probability for a user to have containing a topic in their embedding and the topic distribution for the ML-100k dataset

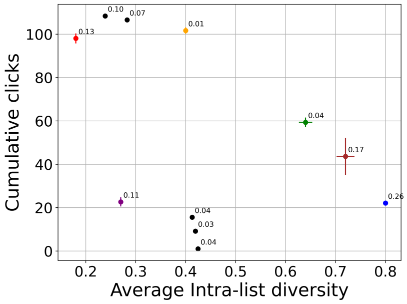
B Multi-Objective Optimization Performance on all slate size - catalog size configurations (RQ1)



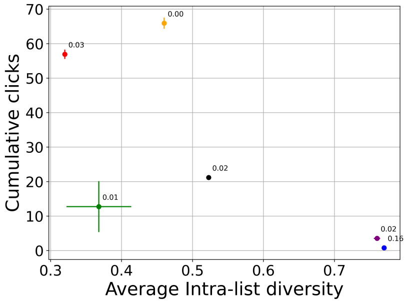
(a) Diversity vs Clicks by Agents for Slate Size 3 and 100 Items



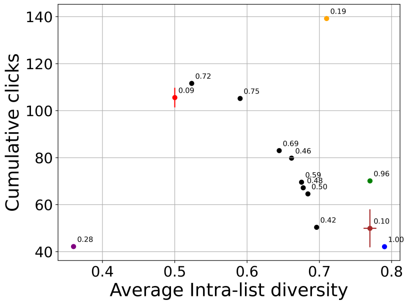
(b) Diversity vs Clicks by Agents for Slate Size 3 and 500 Items



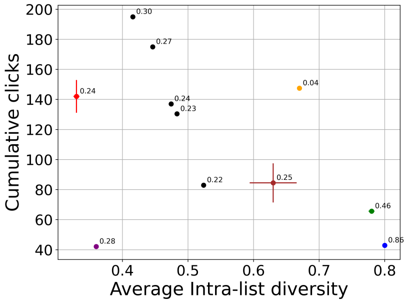
(c) Diversity vs Clicks by Agents for Slate Size 3 and 1000 Items



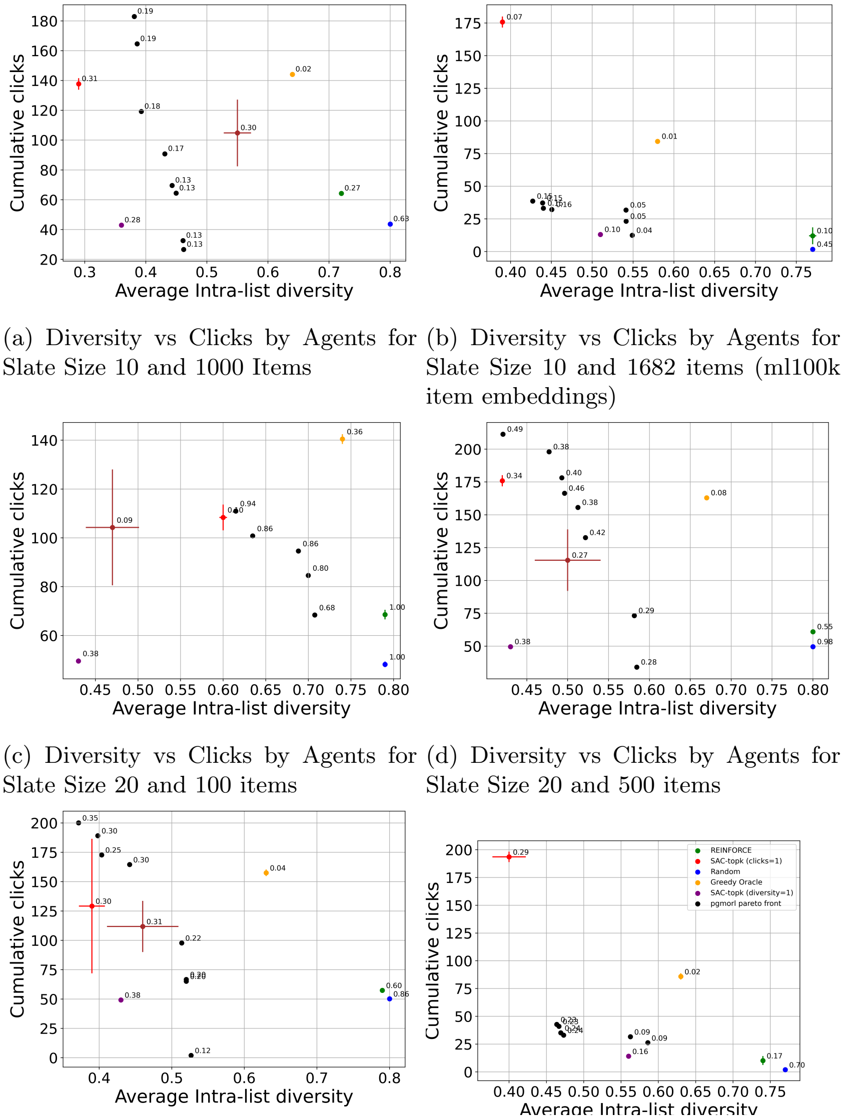
(d) Diversity vs Clicks by Agents for Slate Size 3 and 1682 items (ml100k item embeddings)



(e) Diversity vs Clicks by Agents for Slate Size 10 and 100 Items



(f) Diversity vs Clicks by Agents for Slate Size 10 and 500 Items



(a) Diversity vs Clicks by Agents for Slate Size 10 and 1000 Items

(b) Diversity vs Clicks by Agents for Slate Size 10 and 1682 items (ml100k item embeddings)

(c) Diversity vs Clicks by Agents for Slate Size 20 and 100 items

(d) Diversity vs Clicks by Agents for Slate Size 20 and 500 items

(e) Diversity vs Clicks by Agents for Slate Size 20 and 1000 items

(f) Diversity vs Clicks by Agents for Slate Size 20 and 1682 items (ml100k item embeddings)