Personalized Recommender Systems for Gym Workouts: A Reinforcement Learning Approach

MSc Computer Science - Artificial Intelligence Track

Roan Rosema

Delft University of Technology

TU Delft

Personalized Recommender Systems for Gym Workouts: A Reinforcement Learning Approach

by

Roan Rosema

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on June 1st 2026 at 14:00.

Student Number: 6079598
Project Duration: September, 2025 - June, 2026
Supervisor: Masoud Mansoury
Supervisor: Helma Torkamaan
Faculty: Electrical Engineering, Mathematics and Computer Science, Delft

Cover: Gym at Hotel La Passage in Cairo, Egypt
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

Before starting my Master's in Computer Science at TU Delft, I had already decided to spread my courses over two years and reserve a third year to fully focus on my thesis. I chose the Artificial Intelligence (AI) track, which quickly sparked my interest. After completing most of my core courses in the first year, I used the second year to explore different areas within AI and discover what interested me most. This eventually led me to Natural Language Processing (NLP) and Information Retrieval (IR), two areas that became some of my favorite parts of the program.

Throughout these years, I had always felt somewhat intimidated by the idea of writing a thesis. I knew that I did not necessarily see myself as someone who enjoyed doing research, and the thought of working on one topic for such a long period of time felt a bit scary. Still, I was determined to find a topic that genuinely interested me and that I would be motivated to work on during the final year of my studies. With my interests in NLP and IR in mind, I started looking into different research groups. Recommender systems caught my attention. I had briefly encountered the topic during an Information Retrieval lecture, but I had not explored it in depth before. When thinking about possible directions within recommender systems, I realized that my personal interest in working out at the gym could be combined with this research area. I drafted an initial idea, contacted the research group, and was given the opportunity to discuss it further. That idea eventually became the basis for this thesis.

Looking back, I have really enjoyed the process of working on this thesis. Of course, not every week was equally easy, and there were moments of doubt and frustration, but the overall experience has been very positive. I learned a great deal, both academically and personally, and I am proud of the progress I made and of the thesis that resulted from it. Especially since this thesis uses reinforcement learning, considering that I failed the Deep Reinforcement Learning exam thrice.

First and foremost, I want to thank my supervisors, Masoud and Helma, for their excellent guidance, support, and enthusiasm throughout this entire process. I could not have wished for better supervisors. Our weekly meetings were incredibly helpful and gave me clear direction whenever I needed it. I often felt like I was collaborating with both of you rather than being supervised, which made the process both enjoyable and motivating. Your feedback helped me improve the thesis step by step, and the clear communication throughout the project made it easy to ask questions, discuss ideas, and keep moving forward. Thank you both for your time, patience, and trust!

I would also like to thank my girlfriend Julie for always being by my side. You have helped me tremendously during my time here in Delft, both personally and academically. Life has not been the easiest, and I hope we can continue to support each other through thick and thin. I love you.

Furthermore, I would like to thank my parents, my sister, my brother, my friends, Julie's family and friends, my fellow students, and everyone else who has supported me throughout this period. Whether through advice, encouragement, distraction, or simply being there, you all contributed in your own way. Thank you, from the bottom of my heart.

With this thesis completed, I will now take some time off to explore the world and see where life takes me next.

Carpe diem!

Roan Rosema
Delft, May 2026

Abstract

A good workout is more than a list of exercises. In the gym, recommendations must also decide how much work a user should do, whether that workload is realistic, and how the next recommendation should adapt when a user starts skipping exercises. This makes gym workout recommendation a sequential decision problem rather than a standard item-ranking task.

This thesis studies whether reinforcement learning (RL) improves workout recommendation when the problem is extended from exercise selection to full prescription. Starting from the Home-Fitness RL framework of Tragos et al. [74], we develop a simulator-based gym recommendation framework with four environments: exercise-only and full-prescription settings, each with and without skip-based interaction. The full-prescription environments recommend exercise, sets, repetitions, and load, while the skip-enabled environments use skip-only feedback for online personalization. Because suitable real-world gym interaction data was not available, synthetic user pools were used for training and evaluation under static, dynamic, and stress-test conditions.

The results show that the value of reinforcement learning depends strongly on the structure of the problem. In the exercise-only setting, the RL algorithm Proximal Policy Optimization (PPO) clearly outperforms random recommendation and remains competitive with Particle Swarm Optimization (PSO), but it does not outperform a strong greedy baseline. In the full-prescription setting, PPO becomes the strongest method and outperforms all baselines, showing that reinforcement learning becomes more useful once the recommendation task includes dose and user-specific capacity. Skip-enabled environments lead to more adherence-aware behavior, but also introduce trade-offs between completion and other reward components. Finally, PPO remains stable under realistic gradual user drift, while highly chaotic user changes substantially reduce performance, especially when online personalization is involved.

Overall, the thesis shows that reinforcement learning is not uniformly superior for workout recommendation, but becomes clearly more convincing when the problem is extended to realistic gym prescription and user interaction.

Contents

Nomenclature

This chapter lists the main abbreviations and symbols used throughout the thesis, especially in the Methodology and Experimental Setup chapters.

Abbreviations

Abbreviation	Definition
CI	Confidence interval
EMA	Exponential moving average
HFRL	Home-Fitness Reinforcement Learning framework
IJCAI	International Joint Conference on Artificial Intelligence
MDP	Markov Decision Process
MultiDiscrete	Gymnasium multi-discrete action space
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
1RM	One-repetition maximum
BMI	Body mass index
PA	Physical activity
EONS	`ExerciseOnlyNoSkip`
EOS	`ExerciseOnlySkip`
FPNS	`FullPrescriptionNoSkip`
FPS	`FullPrescriptionSkip`

Symbols

Symbol	Definition	Unit
$\mathcal{X}$	Observation space (observations $o_t \in \mathcal{X}$)	[–]
$\mathcal{A}$	Action space (actions $a_t \in \mathcal{A}$)	[–]
$\mathcal{P}$	Transition dynamics (state transition distribution)	[–]
s_t	Environment hidden state at step t	[–]
o_t	Observation vector at step t	[–]
a_t	Action at step t	[–]
t	Step index within an episode/session ($t \in \{1, \dots, T\}$)	[–]
T	Session length (fixed number of recommendation steps per episode)	[steps]
τ	Episode trajectory, e.g. $\tau = (o_1, a_1, \dots, o_T, a_T, o_{T+1})$	[–]
r_t	Reward at step t	[–]
$R(\tau)$	Terminal session-level reward of trajectory τ	[–]
$\pi_\theta(\cdot \mid o_t)$	Stochastic policy with parameters θ	[–]
θ	Policy parameters	[–]
$J(\pi_\theta)$	Expected discounted return / RL objective	[–]
γ	Discount factor	[–]
$\mathcal{E}$	Exercise catalog index set	[–]
N	Number of exercises in the catalog	[exercises]

Symbol	Definition	Unit
e_t	Exercise index selected at step t	[–]
e_i	Exercise index at session position i	[–]
$\hat{S}$	Suggested session sequence	[–]
S	Completed session sequence	[–]
S^-	Previous session used for comparison (context-dependent)	[–]
$\mathcal{M}(e)$	Muscle set associated with exercise e	[–]
$\mathcal{M}_{\text{all}}$	Global set of all muscle labels across the catalog	[–]
$M(S)$	Muscle set induced by session S	[–]
$\text{routine}(e)$	Routine label of exercise e	[–]
$r^\star$	Sampled target routine for the current session	[–]
W_t	Accumulated workload proxy within a session at step t	[–]
c	Completion ratio, $c = \lvert S\rvert/T$	[–]
$\mathcal{K}$	Set of reward components	[–]
R_k	Reward component k	[–]
$\tilde{R}_k$	Thresholded / penalized reward component k	[–]
w_k	Weight of reward component k in the weighted sum	[–]
τ_k	Penalty threshold for reward component k	[–]
R_{total}	Weighted and clipped total reward before completion scaling	[–]
R_{final}	Final terminal reward after adherence scaling	[–]
w_{decay}	Distance-based decay factor in diversity / fatigue computations	[–]
$\text{sim}(\cdot,\cdot)$	Exercise-level similarity function	[–]
$\text{sim}_{\text{set}}(S, S^-)$	Set-based similarity between two sessions	[–]
p	Consecutive-repeat penalty term in intra-session diversity	[–]
$F(m)$	Fatigue accumulation score for muscle m	[–]
$F_{\max}$	Maximum fatigue score over muscles	[–]
w_i	Muscle-hit weight in fatigue accumulation (1.0 for target, reduced for secondary)	[–]
τ_{fat}	Decay parameter in the fatigue reward	[–]
R_{unique}	Uniqueness reward	[–]
R_{intra}	Intra-session diversity reward	[–]
R_{inter}	Inter-session diversity reward	[–]
R_{routine}	Routine-match reward	[–]
R_{fatigue}	Fatigue reward	[–]
R_{rx}	Prescription quality reward	[–]
R_{persist}	Persistence reward	[–]
b_t^{set}	Sets-bin index selected at step t	[–]
b_t^{rep}	Reps-bin index selected at step t	[–]
b_t^{load}	Load-bin index selected at step t	[–]
$\mathcal{B}_{\text{set}}$	Set of available sets bins	[sets]
$\mathcal{B}_{\text{rep}}$	Set of available reps bins	[reps]
$\mathcal{B}_{\text{load}}$	Set of available load bins	[–]
$\mathcal{W}_{\text{pop}}(e)$	Set of popular $(\text{sets}, \text{reps})$ prescriptions for exercise e	[–]
w	Working weight in the Epley relation	[kg]
1RM	One-repetition maximum strength value	[kg]
1RM_{base}	Baseline 1RM from StrengthLevel	[kg]
$1\text{RM}_{\text{expected}}$	Expected 1RM for a user	[kg]
$1\text{RM}_{\text{chosen}}$	Implied 1RM from chosen load and reps	[kg]
σ_{sets}	Gaussian width for sets similarity	[–]
σ_{reps}	Gaussian width for reps similarity	[–]
σ_{load}	Gaussian width for load similarity	[–]
ρ	Ratio between chosen and target normalized intensity in load scoring	[–]
q_{sr}	Sets+reps plausibility score	[–]

Symbol	Definition	Unit
q_{load}	Load appropriateness score	[–]
q_i	Per-step prescription quality score	[–]
$\bar{q}$	Mean prescription score across a session	[–]
$\text{softmin}_\beta(\cdot)$	Soft-min aggregation over prescription scores	[–]
β	Soft-min sharpness parameter	[–]
w_{load}	Weight of the load term in prescription quality	[–]
w_{sr}	Weight of the sets+reps term in prescription quality / persistence	[–]
α	Recovery scaling factor for repeated same-routine sessions	[–]
$E(S)$	Set of exercises appearing in session S	[–]
R_{ex}	Exercise-persistence sub-score	[–]
$R_{\text{sr}}(e)$	Sets+reps persistence score for shared exercise e	[–]
w_{ex}	Weight of the exercise-persistence term	[–]
Δsets	Difference in mean sets between two same-routine sessions	[sets]
Δreps	Difference in mean reps between two same-routine sessions	[reps]
$p_{\text{true},t}$	True simulator skip probability at step t	[–]
$p_{\text{hat},t}$	Predicted skip probability at step t	[–]
y_t	Skip outcome at step t (1 if skipped, 0 otherwise)	[–]
C_{true}	User's latent true capacity parameter	[–]
$\hat{C}$	Online estimate of user capacity	[–]
b_{hat}	Online skip-bias estimate	[–]
$a_{\text{hat}}(m)$	Online per-muscle avoidance estimate for muscle m	[–]
η_b	Learning rate for skip-bias updates	[–]
η_C	Learning rate for capacity updates	[–]
η_m	Learning rate for per-muscle avoidance updates	[–]

1 Introduction

Regular physical activity is associated with better health outcomes across the lifespan, and current public-health guidelines recommend both aerobic activity and muscle-strengthening activity as part of a healthy lifestyle [78]. In the context of gym training, however, deciding what a person should do is not straightforward. A useful recommendation is about which exercise to perform, but also about how that exercise should be prescribed through sets, repetitions, load, rest, and progression. These choices depend on the user's goals, training status, and current capacity [4].

This makes gym workout recommendation a relevant but difficult personalization problem. Digital fitness tools already collect a large amount of behavioral and physiological information, such as heart rate, pace, Global Positioning System (GPS), and activity metadata, which makes data-driven support increasingly possible [54]. At the same time, it is well known that digital health systems often struggle with long-term engagement and dropout [19]. For workout recommendation, this means that a system should recommend sessions that both look good in theory and that users are willing and able to complete.

Recommender systems have traditionally been studied in domains such as entertainment and e-commerce, where the recommendation task is often framed as selecting or ranking items. However, there is increasing interest in applying recommender systems to health, sports, and physical activity, where the recommendation problem is more structured and unfolds over time [64, 65]. In the fitness domain, existing work has shown that generic or static recommendations can be limited when they do not properly account for user context, constraints, or changing needs [34].

Workout recommendation is especially different from standard recommendation because the object being recommended is usually not a single item, but a session. A workout is naturally sequential: it consists of multiple exercises, and the suitability of each next exercise depends on what has already happened earlier in the session. Exercises interact through factors such as muscle overlap, fatigue, repetition, and session structure [4, 74]. In a gym setting, this becomes even more complex, because recommending only exercises is often not enough. A realistic gym recommendation should also include prescription variables such as sets, reps, and load. This makes the decision problem substantially richer and increases the importance of user-specific capacity.

1.1. Problem definition

In this thesis, gym workout recommendation is defined as a sequential decision problem in which a system constructs a workout session step by step under physiological and behavioral constraints. In the simpler setting, the system recommends exercises. In the richer setting, it recommends both exercises and prescription variables such as sets, repetitions, and load. The quality of each recommendation depends on the current choice, the session history, accumulated workload, user-specific capacity, and whether the user is likely to complete the proposed workout. The central problem is therefore sequential session construction under adherence and prescription constraints, rather than standard item ranking.

1.2. Research gap and thesis objective

Reinforcement learning (RL) is a suitable framework for recommendation problems in which decisions are made sequentially and the quality of the outcome depends on the full sequence rather than on isolated choices. This is relevant in workout recommendation, where the quality of a full session matters more than the quality of one individual recommendation.

However, an important research gap remains. Existing work has shown that RL can be used for sequential home-fitness exercise recommendation, but prior work does not clearly address a realistic gym setting in which recommendations include not only exercise choice, but also full prescription variables such as sets, repetitions, and load. In addition, suitable large-scale gym interaction data is not readily available, which makes direct training and evaluation difficult. It also remains unclear under which conditions RL truly outperforms strong non-RL baselines once realistic user interaction and adherence effects are taken into account.

To address this gap, this thesis builds on the Home-Fitness RL (HFRL) framework of Tragos et al. [74] and extends it toward a more realistic gym setting. The first extension is domain-related: the setting moves from home bodyweight exercise recommendation to gym workout recommendation. The second extension is action-space related: the recommendation problem is expanded from exercise-only recommendation to full prescription recommendation, where the system recommends exercise, sets, repetitions, and load. The third extension is interaction-related: user interaction is modeled more explicitly through skip-only feedback and online personalization. The goal of these extensions is not simply to make the problem larger, but to study more clearly when RL becomes useful compared with strong non-RL baselines.

1.3. Research questions

The main research question of this thesis is:

- ***To what extent does reinforcement learning improve sequential workout recommendations when the domain is extended from exercise selection to full prescription and realistic user interactions?***

To answer this main question, the thesis addresses four sub-research questions:

- **Sub-RQ1:** ***When reproducing the HFRL framework in a gym exercise-selection setting, how do RL policies compare to greedy heuristic baselines, PSO, and random policies?***
- **Sub-RQ2:** ***Does expanding the action space from exercise-only to full prescription (exercise, sets, reps, load) create conditions where RL reliably outperforms non-RL baselines?***
- **Sub-RQ3:** ***How does modeling skipping and online personalization from skip-only feedback affect learning stability and recommendation quality?***
- **Sub-RQ4:** ***Can policies trained with a dynamic user pool match performance of static-pool training, and what changes in learning curves indicate robustness limits?***

1.4. Main findings and contributions

This thesis makes three main contributions. First, it extends prior reinforcement-learning-based workout recommendation from home-fitness exercise sequencing to a gym setting with both exercise-only and full-prescription recommendation environments. Second, it introduces skip-enabled environments that model skip-only feedback and online personalization during session construction. Third, it provides a controlled comparison between RL and non-RL baselines under static, dynamic, and stress-test user conditions.

The results show that reinforcement learning is not uniformly superior across all versions of the problem. In the exercise-only setting, PPO clearly outperforms random recommendation and remains competitive with PSO, but it does not outperform a strong greedy baseline. In the full-prescription setting, PPO becomes the strongest method and outperforms all baselines, indicating that RL becomes more useful once the recommendation problem includes dose and user-specific capacity. Skip-enabled environments lead to more adherence-aware behavior, but also introduce trade-offs with other reward

components. Finally, PPO remains stable under gradual user drift, while highly chaotic user changes substantially reduce performance, especially when online personalization is involved.

1.5. Thesis outline

Chapter 2 reviews related work in physical activity recommender systems, fitness feedback and coaching technologies, and sequential recommendation methods. Chapter 3 introduces the main background concepts and formalizes workout recommendation as a sequential decision problem. Chapter 4 presents the methodology, including the environment design, reward formulation, and simulation process. Chapter 5 describes the experimental setup. Chapters 6 and 7 present and discuss the results. Finally, Chapter 8 concludes the thesis and outlines limitations and future work.

2

Related Work

Research on recommender systems for exercise in the gym remains a largely unexplored research field. In this chapter, we review the previous literature concerning recommender systems that have been utilized in correspondence to various physical activities.

To support reproducibility, the database-specific search strings used for the literature search are included in Appendix A.

2.1. Physical Activity Recommender Systems

Personalized physical activity (PA) recommender systems sit perfectly between standard recommender system methods, sensing, and behavior change. The domain is highly relevant in this day and age as phones and wearables already track a lot of information, including heart rate, pace, location data (GPS) and accelerometry data. This data is both useful for logging and guidance. The task at hand is not to pick an item, but rather to specify a session. What modality, what intensity, how long, under which constraints, and with what goal in sight are all factors that need to be taken into account [65, 64]. The focus in these recommendations is about adherence, safe progression, and wellbeing; both within-session and over time [65].

In terms of representation, physical activities are context-dependent sessions rather than simple products. Attributes include modality, duration, intensity, muscle focus, equipment and environmental constraints. The recommendation often becomes a sequence that encodes the inter-session balance, as well as models the recovery needs rather than a single isolated choice [65]. The personalization of these recommendations must operate under physiological and contextual constraints. These designs move from a simple similarity search to a safety-aware decision support system [64]. In addition to learned user preferences, useful recommendations should respect the readiness, load, fatigue, and safety considerations. Furthermore, evaluation metrics need to acknowledge that the suggestions produced by the system affect bodies and behavior. Offline accuracy alone is often insufficient, hence longitudinal studies are frequently employed [64, 34].

Content-based and user-based collaborative filtering still form sensible baselines when paired with physical activity domain features. Recent work compares these in the wild while keeping the catalog and the UI constant. The effect of ranking logic on both objective and perceived outcomes is thereby isolated. The goal is to measure not just the accuracy, but also the inspiringness, diversity, and motivational impact over multiple weeks. It models the within-person change rather than relying on only offline metrics [12]. The main conclusion from such designs is not that the accuracy is completely irrelevant, but that behavior and experience are the main signals when recommendations are meant to structure your everyday activities [12].

On the data side, recommender systems in the fitness domain often combine sensory data, such as heart rate, GPS, inertial measurement unit (IMU) sensor data, with relatively simple in-app feedback, such as ratings, novelty and motivation. Records show that generic tracker suggestions tend to un-

derperform when they fail to account for a person's current state, intent, and constraints [60]. This strengthens the case for context-aware and sequential approaches, and it motivates the use of models that capture a person's dynamics rather than treating the interactions between the system and the user as independent observations [34, 12].

The main takeaway from these foundations is to treat PA recommendations as decision support under constraints, where sessions should be viewed as context-dependent and plans should be made sequentially. Recommendations must be safe and actionable by using clear explanations and providing recovery guidance [65, 64, 12].

2.2. Core Recommendation Paradigms for Physical Activity

2.2.1. Context-aware and Social PA Recommenders

Context-aware PA recommender systems take the approach of giving recommendations where the value of the suggestion depends just as much on the situation as on the user's preference. The same user may value the suggestion of a nice run outdoors on a sunny Saturday as much as the suggestion of doing a small workout at home on a rainy day after working all day. An early 2014 recommender system known as RecFit makes this dependence tangible by enriching a catalog of activities with contextual attributes that can be directly evaluated at recommendation time. These attributes include factors such as intensity, cost, injury risk, sociability, location, equipment requirements, time windows, and weather suitability. The system filters and ranks the possible activities to produce a shortlist that fits the user's current constraints [40]. Although not making use of standard recommender system methods, this design of adding contextual attributes to activities matters as it supports hard filters and soft utilities, and yields recommendation logic that users can understand and negotiate.

The same idea is extended by systems using phone sensory data, where the context is sensed as opposed to being declared. In the smartphone PA recommender by Kadri et al., accelerometer streams are segmented and classified into basic behaviors such as walking, jogging, walking up the stairs, sitting, and standing [43]. The retrieved data is converted into simple health references such as daily energy balance and time in moderate-to-vigorous activity. The recommendations are then framed in terms of the feasibility right now rather than future goals. The study reports a relatively simple decision tree model which performs well on its dataset, but its main contribution comes from the loop of sensing, recognizing, suggesting and collecting feedback.

Social information functions as an additional context layer. Social recommendation for personalized fitness assistance by Dharia et al. supports the behavior and preference of peers or 'near-neighbors' to find approachable activities, which is especially helpful for novice users that benefit from closely related user examples [13]. Social signals can also support safety and adherence by steering users toward group activities at times when self-motivation is low, or by suggesting locally feasible options (e.g. nearby open facilities), in turn aligning opportunity with motivation.

Finally, exercise recommendations for older individuals show that context also includes the capability and risk profiles that change with age. Systems designed for middle-aged and elderly users must remember the lower intensity bands, the risk of falling, and recovery windows. These recommender systems tend to favor simplified choices with conservative defaults. Taking the demographic into account is a contextual attribute that is not only a personalization nicety, but a safety requirement ensuring recommendations to remain cautious [80].

2.2.2. Sequential Planning and Session Composition

As opposed to context-aware recommenders that optimize a single decision, sequential recommenders optimize progression. The unit of choice is a context-dependent session, and the recommendation object is a sequence that respects the dynamics constantly changing within a session, such as fatigue and safety. Safety and recovery are part of the objective, not external constraints. The reality of the data drives the modeling choices, and the evaluation should preferably be longitudinal. Sequence-aware PA recommenders manage adaptation by learning a policy that composes sessions under multi-objective rewards or by predicting the next activity and success modeling [74, 5].

Early gym-related architectures such as the AI Fitness Assistance System (FAS) predict the exercise,

sets, reps, and rest times for a session using artificial neural networks and logistic regression. Simple reinforcement learning schemes are used to explore alternatives across sessions. Even where the modeling details are minimal, the architecture signals a move from item selection to policy control over time [75].

A more data-driven approach appears in Building a Personalized Fitness Recommendation Application based on Sequential Information, where a practical pipeline is realized [2]. It ingests wearable traces and metadata, preprocesses and feature engineers the sequential heart rate (HR) patterns, and utilizes historical patterns to recommend the next physical activity. This idea is important because deployment usually relies on good data handling and iterative evaluation as opposed to a single algorithm. It also aligns with the contextual representations of activities, since those attributes help regularize sequence models.

Modeling both next activity and completion likelihood raises the practical value of sequence models. The Interconnected Recurrent Neural Networks formulation combines association-rule mining, embeddings, and two coupled recurrent networks. One network predicts the next activity and the other models the probability the activity will be completed successfully, changing the recommendation from 'what comes next?' to 'what comes next that the user will actually do?' [46]. This distinction is important in the physical activity domain, where adherence is not only an evaluation metric but rather a constraint. An optimal plan that a user will not follow is not an optimal plan, calling for tangibility in recommender systems.

Reinforcement learning makes the objective trade-offs a main focus. In the home-fitness reinforcement learning framework the problem is cast as episodic policy learning, where each episode is a workout session composed of up to n_s bodyweight exercises. The agent recommends one exercise at a time until the session ends, with the objective to maximize user satisfaction grounded in sports science metrics [74]. The system optimizes a reward that blends goal alignment, intra-/inter-session diversity, fatigue penalties, and lightweight user feedback into a single policy objective. To mitigate data scarcity and the cold start problem, the policy is pre-trained against a user simulator fitted on probabilities of logs collected via a small user study. The system's choices make constraints and explanations straightforward, while the policy structure naturally supports the sequence-level control over the various factors of the reward design. Using the reward design of this system, the policy can be tuned to context and person without needing manual heuristics.

Crossing over to another system that uses reinforcement learning, PERFECT adopts a contextual bandit for walking prescriptions that responds to live heart rate zone feedback [5]. When HR exceeds certain thresholds, the system suggests pausing or adjusting the activity at hand. It uses mixed-effect models to evaluate changes in light to moderate activity minutes over 12 weeks. The system runs on both phones and wearables (smartwatches) and wraps the physiological safety directly into the mechanism. The bandit offers a middle ground between one-shot ranking and full reinforcement learning.

2.2.3. Exploration vs. Repetition

The right balance between exploration and repetition in physical activity recommendations is often dynamic and context-dependent. The trade-off between recommending a novel activity and an activity performed earlier is a common problem that has been studied in various cases. One study used a micro-randomized trial, where the outcome shows that exploration tends to be preferred overall, although the effect depends on the recommended activity type (e.g., general activity vs. intensive activity) and evolves over time. After two weeks, a notable shift is found towards the preferability of repetitious recommendations [9].

A new analysis by the same authors treats the decision of the recommendation with respect to repetition and exploration as a predictive problem and explains it with feature importance and SHAP (SHapley Additive exPlanations) values. The main contextual drivers include the season, hour, temperature, PA situation (location), day of week, and PA type. In the early phases, the users prefer new recommendations, but a shift towards repetition emerges after 15 days, which is in line with the results of the first study. Further results include a bias towards new activities on Sundays, and users in the trial conducted in spring are more likely to show larger weekly gains in doing moderate-to-vigorous activities compared to users of the trial conducted in autumn [11]. A static exploration rate is suboptimal. The

results suggest to start with recommending newer items, and after some time shift into repetition. When context affords variety (e.g., sunny weather or different schedules), exploration should once again be considered.

The studies recommend treating exploration as a main feature instead of a side effect, but instead of fixing it as a constant parameter we should let it vary with context and time. Novelty, inspiringness, and diversity should be logged alongside adherence so the policy can respond before boredom or fatigue appears, combining this with short explanations to justify the suggestion to enhance trust in the system (”new route today because of the rainy weather and your recent preference of outdoor runs”) [10, 11, 9].

2.3. Coaching, Form Quality and Real-Time Feedback

2.3.1. Wearables and Classic Coaching

Classic wearable coaching establishes the practical baseline for real-time interaction and creates the data traces that later, more advanced feedback models can build upon. Early real-time feedback coaching systems such as FitCoach treat the wearable and phone as both a sensor and an interface. They aim to recognize what the user is doing and to convert signals like accelerometry data, gyroscope traces, and heart rate into coaching suggestions [35]. FitCoach performs continuous monitoring of common strength and exercise activities, set and rep counting, exercise recognition, and session summarization, with the goal of providing direct feedback and guidance as opposed to an extensive afterward analysis. The models are deliberately simple, as the suggestions need to be recommended in real-time. These systems matter as a form of verification layer that a plan-level recommendation depends on. If the app can reliably tell what the user has done, then it can decide whether to repeat it, progress, or switch. It can do so based on evidence rather than assumptions. Suggestions like ”slow down when lowering” are easier to follow than an ambiguous score, denoting the most useful feedback is also the simplest. Making the detection robust to typical conditions like wrist wobbling, phone in pocket, or noisy heart rate sensor data is often of greater user value than a classifier that does not succeed well outside the lab [35].

2.3.2. Computer Vision/3D Pose and Physics-Based Feedback

Where wearable coaching verifies that work has occurred, vision-based systems attempt to assess how well the work was performed and to translate that into human-understandable feedback. AIFit is one of the earlier systems that pairs 2D and 3D pose estimation with models producing textual guidance about form. By converting complex model feedback into human-understandable texts (range of motion, joint angles, etc.), the system provides corrections important for the user such as ”deepen the squat until upper legs are parallel to the ground”, which users can act upon without expert supervision. Accuracy alone is not sufficient if the system cannot express what to change and why, and AIFit tries to keep the explanation layer close to concepts like force and motion, so that the guidance remains credible in everyday training contexts [27].

A different approach brings learned physics into play to reason about forces, constraints, and temporal dynamics that are hard to capture with only pose heuristics. In Learnable Physics for Real-Time Exercise Form Recommendations, interaction-network style models use tracked keypoints to approximate the underlying mechanics and produce assessments that can be suggested during a repetition. Not only is better accuracy the main promise, but better yet is the ability to detect errors instantaneously providing actionable suggestions before the errors become too ingrained [42].

These systems based around user safety is not only used in the fitness domain, but the same idea is pushed into other specific risk domains. Lift It Up Right formulates lifting as a posture-quality problem, where pose features are combined with a Lifting Index score to recommend safe adjustments. In some cases, the suggestion proposes alternative variations that reduce spinal load or the risk of shoulder injuries [15]. Rather than aiming for a generic form score, the model prioritizes error classes that matter for injury prevention, so that the recommendations can be embedded into everyday training without the need of expert supervision. This focus yields clear thresholds and provides explicit explanations, which also helps with user trust and compliance.

Recent work also explores whether large multimodal models (LMMs) can provide great pattern recognition and give flexible error descriptions from short videos. Fine-Tuning LMMs for Fitness Action Quality Assessment adapts a video-language model to classify the exercise type, identify the error categories, and temporally localize where the error occurs [14]. The system also extends the prediction with textual rationales. However, calibration, data coverage, and the risk of over-confident feedback remain issues that could be critical in the performance of these systems. The most credible path is to hybridize, using various methods together for a more robust system [27, 42, 15, 14].

2.4. Endurance and Marathon Recommenders

In the context of physical activity recommender systems, multiple studies have been conducted in the endurance and marathon domain. The objective is less about recommending an isolated workout, but rather guiding a runner toward a concrete goal such as finishing the marathon, hitting a target pace, or achieving a personal best (PB). Early systems already treat the marathon as a sequence-design task with safety and feasibility constraints. They combine personal profiles with historical performances to predict challenging but achievable PB finish time and recommend a pacing plan over 5km segments [66, 67]. These systems can meaningfully affect adherence and outcomes by aligning recommendations with the demands of long marathons or events by taking energy management and fatigue into account [47].

In-race guidance treats pacing as a real-time forecasting and control problem. Pace My Race exemplifies this approach by predicting finish time and recommending pace adjustments as the users' conditions and runner state evolve. By treating the marathon as a sequence of decisions rather than a single target, the system aims to avoid starting too fast, 'hitting the wall', or overcorrecting after a bad split time, and to keep the runner on a plausible trajectory. Stable guidance can be delivered during the race by feature engineering over time series data combined with robust models [8, 22].

A separate thread studies who finishes successfully and which training patterns correlate with completion. The collaborative filtering (CF) approach to completing the marathon uses similarities between runners to recommend target times or plan adjustments that increase the likelihood of finishing a run rather than quitting late in the race [6]. Using CF is a logical method, as new recreational runners often face data sparsity at the individual level. A neighborhood of comparable runners compensates for little personal history and provides corresponding guidance.

Fit to Run argues that in-race success depends on how well the plan captured the athlete's context and respected the progression principles, by looking at the training-plan personalization [7]. The system selects and adapts plan components to fit the runner's profile and schedule, then monitors adherence to decide when to progress or hold. This approach treats plan composition as a recommendation problem under constraints, bringing the same logic from Sections 2.1-2.2 into the endurance and marathon domain [23, 24, 26].

Other works in this domain emphasize the use of case-based reasoning (CBR) to extend the same goals across planning, training and explanation. CBR has been used to plan marathon race strategies and predict realistic race times [67, 22], to adapt pacing to ultra-marathon events [47], and to provide explainable projections and training plans in real-world training logs [47, 25, 23]. Other works expand the scope to injury-aware systems and interactive tools that let expert coaches shape the model's emphasis, concurrently improving trust and accuracy [21, 20, 49], as well as generate targeted training adjustments for different finishing time ambitions and evaluate them for feasibility [24].

2.5. Human Factors: Motivation, Persuasion and Privacy/Control

2.5.1. Engagement and Motivation

In the domain of physical activities, a technically 'good' recommendation fails if it does not sustain engagement. Early persuasive eHealth work shows that monitoring how often a user responds, or how their effort changes, can signal declining motivation early enough to intervene with simpler tasks or adjusted goals [55]. The system should detect when recommendations are starting to miss, explain why a change is suggested, and offer a credible next step that preserves the habit of performing activities. Perceived fit, novelty and, inspiringness need logging alongside behavior. Loss of motivation detecting

should be treated as a functionality within recommender systems in this domain [55].

2.5.2. Privacy and Preference Control

Because PA recommenders depend on intimate data (location, heart rate, routines), privacy preferences must shape data collection and use. A recommendation approach by Sanchez et al. in the fitness domain models users' privacy preferences and clusters permissions into usable sub-profiles, which then predicts which profile fits a new user best [61]. Recent systems add privacy by design so less sensitive information leaves the devices in the first place. P3FitRec delivers personalized exercise recommendations while reducing the dependence on sensitive attributes, learning from wearable signals and structuring the model to limit unnecessary exposure to personal details [45].

2.5.3. Call for Tangibility in PA Recommenders

Many recommender systems in the physical activity domain lack a sense of tangibility. Tangible recommendations are physical activity recommendations a user can immediately understand and act on after listening to feedback from the user (either implicit or explicit), because the system makes clear why this is the right suggestion now and what should change after following it [60]. The suggested framework centers on three concerns: trust and interpretation, intent and alignment, and consequence-awareness. Trust grows when the system offers an explanation tied to the user's situation, for example acknowledging fatigue. Intent refers to alignment with what the user actually wants at that moment and the ability to redirect or postpone goals without breaking the plan. Consequence means pairing each suggestion with an expected effect that the user can understand, such as finishing a run without experiencing pain. For example, fatigued runners should not be forced to keep pace, but rather find a new plan adjusted to the possibilities of the user. Cyclists who are recovering from a knee injury should not be recommended the same training plans as before the injury. Evaluation should prioritize lived outcomes such as recovery support, variety, and reflective interaction. Offline accuracy is not enough, recommendations must remain trustworthy and consequence-aware in everyday use [60].

3

Background

This chapter introduces the main concepts needed for the rest of the thesis. It focuses on three topics that are central to this work: workout recommendation as a sequential decision problem, RL as a way to model such problems, and the HFRL Framework proposed by Tragos et al., which forms the main starting point for this thesis [74].

3.1. Workout recommendation as a sequential decision problem

Workout recommendation differs from traditional recommendation because the output is not a single item, but a full session. A recommended workout is a sequence of exercises, and the quality of that sequence depends on how the exercises fit together. Because of this, the problem is naturally sequential.

There are several reasons for this. First, exercises within a session interact through fatigue and recovery. If two exercises train the same muscle groups close together, the second exercise may be less suitable or feel repetitive. The HFRL framework makes this explicit through a muscle-fatigue reward that decreases when the same muscle groups are trained in close succession. Similar ideas also appear in resistance training guidelines, where exercise order, loading, and recovery are treated as important parts of training design rather than as separate details [4].

Second, workout sessions are usually built around goals or routines that apply to the whole session rather than to a single exercise. A user may want a strength-focused session, a cardio-focused session, or a routine-specific session such as push, pull, or legs. In the HFRL framework, this appears through reward components for goal matching, focus-muscle matching, and fitness-level matching. These objectives cannot be handled well by looking only one step ahead, because they depend on the session as a whole.

Third, user interaction creates feedback during the session. A user may skip or replace exercises that feel unsuitable, too difficult, or repetitive. These responses are influenced by what has already happened earlier in the session. For example, skipping may become more likely later in the workout or after several similar exercises. This means that workout recommendation is not just a ranking problem, but also a planning problem under user response.

Taken together, these properties make workout recommendation better described as session construction under constraints than as standard top-N recommendation. A session may look reasonable when each exercise is considered separately, but still be poor overall if it is too repetitive, does not match the intended routine, or becomes too difficult to complete.

3.2. Reinforcement learning for sequential recommendation

Reinforcement learning is a suitable approach for recommendation problems in which decisions are made step-by-step and the quality of the final outcome depends on the full sequence of actions. In the

standard RL setting, an agent interacts with an environment over time, selects actions according to a policy, and tries to maximize expected cumulative reward [73]. This makes RL different from methods that evaluate each recommendation mainly in isolation.

In recommender systems, RL is often used when the system should optimize both immediate user response and longer-term outcomes such as engagement, satisfaction, or retention. Surveys on RL-based recommender systems describe this as the difference between myopic prediction and sequential decision-making, and show how recommendation can be formalized as a Markov decision process when user state changes over time [3].

This is especially relevant in fitness applications, where maintaining engagement is an important challenge. In digital health research, dropout is a common problem, and Eysenbach's "law of attrition" highlights that many users stop using digital interventions before completion [19]. In that sense, adherence is not a secondary issue, but part of the main recommendation objective.

A second challenge is the limited availability of large-scale interaction data. In many recommendation domains, systems can learn from large historical logs. In workout recommendation, and especially in session construction from scratch, such data is often not available. This makes it difficult to train sequential models directly from real interaction traces.

One common solution is to use simulation. Simulation environments make it possible to generate synthetic interaction data, carry out controlled experiments, and train sequential models when real-world logs are limited. RecSim is a well-known example of this idea and was proposed as a configurable platform for studying sequential recommender-user interaction [41].

In this thesis, RL is used in this sequential recommendation sense. The agent constructs a workout session one step at a time, and performance is judged at the session level rather than at the level of individual recommendations.

3.3. The HFRL framework of Tragos et al.

An important starting point for this thesis is the paper "Keeping People Active and Healthy at Home Using a Reinforcement Learning-based Fitness Recommendation Framework" by Tragos et al. (IJCAI 2023), of which we refer to the proposed framework here as the HFRL framework [74]. The framework was developed in the context of increased at-home fitness activity during the COVID-19 period and aims to improve personalization and long-term user engagement through sequential workout recommendation.

The HFRL framework is important for this thesis for two reasons. First, it shows that reinforcement learning can be applied to workout recommendation as a sequential planning task. Second, it provides a concrete design that can be extended toward a more realistic gym setting.

At the same time, the paper does not fully specify every implementation detail needed for exact reproduction. Several parts are described clearly at a conceptual level, but some lower-level design choices remain implicit. For this thesis, the HFRL framework is therefore treated as the main conceptual and methodological starting point, while certain implementation choices had to be interpreted and made explicit in our own setup.

The overall HFRL framework can be seen in Figure 3.1.

3.3.1. Core setup: app context, action space, and sequential episode

In the HFRL framework, the recommendation task is to construct a full workout session by selecting exercises sequentially from a catalog in a mobile fitness application. The paper describes an action space of 161 bodyweight exercises, each with attributes such as difficulty, target muscles, style, and equipment requirements.

The state representation includes user-related information, workout goals, session status, and feedback signals such as likes, dislikes, and whether earlier exercises were skipped or completed. The framework models this problem using the available observable state. The paper also reports that the policy network receives an observation vector of size 1287, but it does not give the exact feature-by-feature

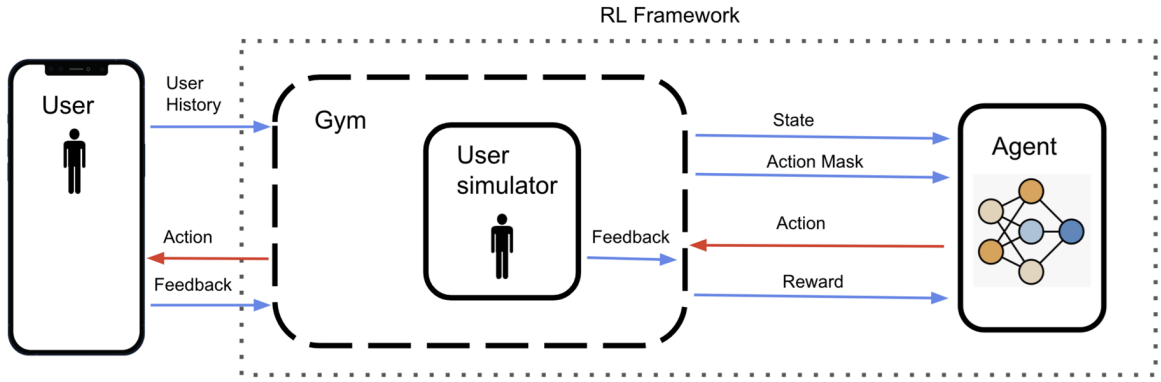


Figure 3.1: The overall structure of the HFRL framework. Image source: [74]

breakdown of this vector in the published version [74]. Likewise, feedback is described conceptually as part of the state, but the exact way in which each signal is encoded into the input is not fully specified in the paper.

The general setup is therefore an RL environment in which the agent recommends one exercise at a time, the environment updates the session state, and reward is returned based on the quality of the final workout.

3.3.2. HFRL's reward design

A central part of the HFRL framework is its reward design. Instead of learning reward from large-scale interaction data, the framework defines several workout-quality metrics and combines them into a session-level reward. The paper includes the following main components:

- Intra-session diversity, based on the similarity of muscles trained by exercises within the session, with additional penalties for consecutive repeats.
- Inter-session diversity, encouraging variation across the last L sessions.
- Fitness-level matching, encouraging exercises whose difficulty is aligned with the user's assessed difficulty preference.
- Goal matching (cardio vs strength), based on the intended workout style.
- Focus-muscle matching, measuring alignment between session exercises and user-selected primary and secondary muscle targets.
- Muscle-fatigue penalty/reward, decreasing when the same muscles are repeatedly loaded in close succession.
- A wrists-related penalty, motivated by user requirements and injury risk concerns for wrist-dependent movements.

These components are intended to capture whether a session is varied, aligned with user goals, and physically appropriate. The paper also applies threshold-based penalization and boosting, so that sessions that violate important criteria receive clearly worse rewards.

Another important choice is that reward is mainly terminal. In other words, the framework returns non-zero reward only at the end of the workout session. This reflects the practical idea that users often judge a session only after completing it, rather than after each individual exercise.

At the same time, not all reward details are fully specified in the paper. The authors state that the final reward is a weighted average of the sub-rewards and that the weights are chosen using domain knowledge, but the exact sub-reward weight values are not reported [74].

3.3.3. User simulator and implicit feedback through skipping/replacement

Because real-world training data was limited, the HFRL framework includes a user simulator that generates synthetic users and interaction behavior [74]. The simulator samples user profiles from probability distributions and assigns attributes such as age, gender, fitness level, goals, and exercise preferences.

The framework also models interaction feedback, including skipping and exercise replacement. The paper explains that these behaviors were informed by an exploratory user trial and were used to estimate probabilities for skipping or changing an exercise during a session. Factors such as dislike, repetition, and later session position increase the chance of skipping.

This is important conceptually because it shows that adherence is part of the recommendation problem itself. A session is both judged by how good it looks structurally and by whether the user is likely to follow it.

However, some simulator details remain unspecified in the published paper. The authors mention that their exploratory analysis resulted in probability equations for skipping and changing exercises, but these equations are not reported explicitly [74] and the source code is unavailable.

3.3.4. Learning algorithm and constraints

For policy learning, the HFRL framework uses PPO. The paper describes a neural-network policy that maps the observation vector to a distribution over exercise actions, together with a value function that shares most of the same architecture.

The framework also applies action masking to remove invalid exercises, such as exercises that require unavailable equipment. This is a practical way of handling constraints inside the action space.

PPO is a common choice in applied deep RL because it is relatively stable and simple to use in larger environments [62]. In the HFRL framework it serves as the main learning method for sequential workout construction.

3.3.5. Baseline and evaluation setting

The HFRL framework compares the RL policy against a Particle Swarm Optimization (PSO) baseline. The paper describes PSO as optimizing session properties such as diversity and fitness alignment, but without the same level of personalization through user feedback.

In simulation, the paper reports that the RL model outperforms PSO in average session score. More importantly, the work also includes a 15-week randomized crossover user trial. In that trial, users alternated between RL-based and baseline recommendations, and session satisfaction was measured using the ”Physical Activity Enjoyment Scale-8” (PACES-8) questionnaire. The paper reports higher mean satisfaction for RL sessions and more positive trends over time [74].

For this thesis, the importance of the HFRL framework is both in its reported results and in its structure. It provides a concrete example of how to formulate workout recommendation as an RL problem, how to define session-quality reward components, and how to compare RL with a non-learning planning baseline.

3.3.6. Implementation uncertainties in the HFRL framework

For this thesis, the HFRL framework was used as the main conceptual starting point. However, while studying the paper in detail, several implementation-related uncertainties remained. These are worth stating explicitly, because they influenced how the current work was designed and how closely exact reproduction could be attempted.

First, the paper reports an observation size of 1287, but does not provide the exact composition of that vector. The paper describes the state at a high level in terms of user information, goals, session status, and feedback, but the precise encoding is not given in the published version [74].

Second, the reward design is conceptually clear, but not fully specified numerically. The paper states that the final reward is a weighted average of the reward components and that the weights are determined by domain knowledge, but the exact weight distribution is not reported [74]. This is important for

reproduction, since different weight choices can materially change the learned behavior.

Third, the role of skipping in the final reward is not fully explicit. The paper clearly models skipping and exercise replacement in the simulator, but it is less clear from the published description whether the reported reward is based on the suggested session, the completed session, or includes some separate adherence penalty beyond the simulator interaction itself. This distinction matters for interpreting how strongly skip behavior affects training.

Fourth, much of the paper's evaluation centers on user satisfaction through PACES-8 and on comparison against a PSO baseline. This is a reasonable evaluation choice, but it also means that some lower-level implementation details are less central in the presentation than the overall empirical result.

Fifth, in the evaluation section it is unclear how many focus muscles were chosen per session. If users select only very few focus muscles, reward scores would generally be lower than sessions where all muscles are chosen, due to the lack of diverse available actions to generate a full session with, without generating muscle fatigue. Without this knowledge, interpreting the results is done with the assumption that a reasonable set of focus muscles are chosen, however, we cannot be certain.

Because of these uncertainties, the HFRL framework was not treated in this thesis as a line-by-line implementation blueprint, but as a strong conceptual and methodological foundation. We also attempted to contact the authors directly with clarification questions about the input representation, feedback encoding, reward weights, simulator design, skip/change probability equations, and the role of focus muscles, but we did not receive a response. For that reason, where the paper was underspecified, explicit implementation choices had to be made in the present work.

3.4. From the HFRL framework to gym recommendation

This thesis builds on the HFRL framework, but extends it to a gym setting and to a richer planning problem. The goal of these extensions is to both make the setting more realistic, and also to study more clearly when reinforcement learning provides an advantage over strong baselines.

3.4.1. Why gym recommendation is a harder planning problem

A home bodyweight session can often be described reasonably well by a sequence of exercises, because intensity is partly limited by bodyweight and by the difficulty of the movement itself. In a gym setting, this is no longer enough. A recommendation must also specify sets, repetitions, and load, and these choices strongly affect both the training stimulus and how difficult the session feels.

Resistance training guidelines make clear that programming depends on training variables such as load, repetitions, sets, rest, frequency, and progression, and that these choices should depend on the user's training status [4]. In practice, this means that choosing the right exercise is only one part of the recommendation problem. The system must also choose a suitable prescription for that exercise.

From an RL perspective, this makes the planning problem harder in several ways. First, the action space becomes much larger, because the system must choose both an exercise and prescription variables. Second, user-specific capacity becomes more important, since the same exercise may be manageable or unmanageable depending on the chosen load and volume. Third, user feedback becomes harder to interpret, because skipping may be caused by both the exercise itself and by the prescription intensity.

For these reasons, gym recommendation is a more demanding setting than the original home-fitness problem. It requires the model to reason about both the session structure and the amount of work assigned in each step.

3.4.2. Why skip-based feedback matters conceptually

The HFRL framework already shows that skipping is an important form of user interaction. In that framework, skip probability depends on context, such as late-session position, dislike, and repetition. This suggests that skipping is not random, but contains useful information about how well the recommendation matches the user.

In recommendation terms, skipping can be viewed as a form of implicit feedback. It is not an explicit rating, but it still reveals something about mismatch between the recommendation and the user's current preference or capacity. In sequential settings, this matters even more, because skip behavior can influence both how past recommendations are evaluated and how the next recommendation should be made.

This is especially relevant in a gym setting. A skipped exercise may mean that the movement itself is unsuitable, but it may also mean that the prescription is too demanding or that the session workload has become too high. Skip behavior therefore gives information about both exercise suitability and current tolerance.

In this thesis, skip-based feedback is used as a source of online personalization. The idea is that skipping can help bridge the gap between sessions that look good on paper and sessions that users are actually willing and able to complete. This makes skipping conceptually important for both evaluation and adaptation during the session.

3.4.3. Summary of the extension direction

In short, the HFRL framework provides a clear starting point for this thesis: sequential exercise recommendation, a user simulator for training under limited data, and reward functions based on session-quality criteria. This thesis keeps that general structure, but extends it in a more realistic gym direction.

The first extension is domain-related: the setting moves from home bodyweight workouts to gym workouts. The second extension is action-space related: instead of recommending only exercises, the full-prescription environments also recommend sets, reps, and load. The third extension is interaction-related: skip behavior is modeled explicitly and used as a source of online personalization through skip-only feedback.

Together, these changes increase both the realism and the difficulty of the planning problem. They also make it possible to study more clearly when reinforcement learning becomes useful compared with non-learning baselines [3].

4

Methodology

This chapter describes the methodology used in this thesis. It explains how the gym recommendation problem is formulated as a reinforcement learning task, how users and exercises are represented, how the simulator is constructed, and how rewards, environments, and evaluation are defined.

4.1. Problem formulation

We model the gym workout recommendation as an episodic reinforcement learning problem. The agent acts as a workout planner that constructs a session exercise-by-exercise, while a user simulator provides interaction dynamics and implicit feedback via skipping. Each episode represents one workout session with a fixed length T.

4.1.1. Episodic RL

The task can be formalized as an episodic Markov Decision Process (MDP) [76] $(\mathcal{X}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where r denotes the reward function. At each step $t \in \{1, ..., T\}$, the agent receives an observation vector o_t (a function of the underlying state s_t), selects an action $a_t \sim \pi_\theta(\cdot \mid o_t)$, and the environment transitions to a new state $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$.

In our setting, the environment includes a user simulator that controls how the user responds to recommendations and how user-related variables evolve over time. Some user features are hidden from the agent, such as a true capacity value that affects skipping. For that reason, the problem is more naturally viewed as a Partially Observable MDP (POMDP). In practice, we treat it as an MDP over the observation space by including enough history and interaction signals in o_t.

4.1.2. Episode definition

An episode is a sequence:

$$\tau = (o_1, a_1, o_2, a_2, ..., o_T, a_T, o_{T+1}), \tag{4.1}$$

representing one workout session. The session terminates after T steps. We use a terminal, session-level reward: intermediate rewards are zero, and the final reward is computed once the full session is constructed:

$$r_t = 0 \quad \text{for } t < T, \qquad r_T = R(\tau). \tag{4.2}$$

4.1.3. Action spaces: exercise-only vs full prescription

We consider two action formulations:

Exercise-only recommendation

The agent selects a single exercise index from a catalog of size N:

$$a_t \in \{0, ..., N-1\}.$$

This corresponds to a discrete action space $\mathcal{A} = \text{Discrete}\{N\}$.

Full prescription recommendation

The agent selects both the exercise, as well as prescription parameters (sets, reps, load), partitioned into bins:

$$a_t = \left(e_t, b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right), \tag{4.3}$$

where $e_t \in \{0, \ldots, N-1\}$ and $\left(b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right)$ are bin indices for sets, reps, and load. This corresponds to a multi-discrete action space $\mathcal{A} = \text{MultiDiscrete}\left([N, |\mathcal{B}_{\text{set}}|, |\mathcal{B}_{\text{rep}}|, |\mathcal{B}_{\text{load}}|]\right)$.

The full-prescription setting increases the complexity of the action space and makes it possible to include reward components that depend on sets, reps, and load. These components are not available in the exercise-only environments.

4.1.4. Observability and hidden state

The recommendation process is partially observable. At each step t, the agent does not observe the full simulator state s_t directly, but instead receives an observation vector o_t. This observation only contains the information that is made available to the policy. The exact construction of o_t is described later in Section 4.2.4.

The underlying state s_t contains additional simulator variables that are hidden from the agent, such as latent user factors affecting skipping behavior and transition dynamics. In particular, user-specific quantities such as true capacity are used internally by the simulator but are not directly exposed to the policy.

4.1.5. Transition dynamics and user interaction

After the agent proposes an action a_t, the environment updates session history and simulates user interaction, depending on the environment variant.

No-skip environments

Each suggested exercise is treated as completed. The transition is mainly keeping track of what happened during a session (history masks, routine context, user history update at the end of an episode).

Skip environments

The user simulator decides whether the suggested exercise is skipped based on a skip probability model. The environment records both the suggested and completed sequences. Skip feedback is also used to update the online estimate of user-specific parameters like the capacity estimate $\hat{C}$ and potential per-muscle avoidance terms. This creates a feedback loop where the agent can adapt its choices based on observed interaction signals within the episode.

4.1.6. Learning objective

The goal is to learn a policy π_θ that maximizes expected return. Because we use a terminal reward, this amounts to maximizing the expected quality of the final session:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\gamma^{T-1} R(\tau)\right]. \tag{4.4}$$

In practice, we optimize π_θ using PPO and evaluate performance using both the overall episode return and the individual reward components, described later in Chapter 4 and 5.

4.2. Data representation

This section describes how workouts, users, and interaction signals are represented in the environments. All environments are built around a shared exercise catalog derived from the ExerciseDB V1 dataset and a synthetic user pool sampled from distributions meant to reflect realistic gym users [18].

4.2.1. Exercise catalog

Let $\mathcal{E} = \{0, ..., N-1\}$ denote the set of available exercises in the catalog, where each exercise $e \in \mathcal{E}$ corresponds to one row in the exercise table. Each row contains metadata used for defining routines, computing reward components, and constructing observation features. Key fields include:

- Routine label (push/pull/legs), used to sample a session routine.
- Target muscles and secondary muscles, stored as strings and parsed into a normalized list of muscle labels.
- StrengthLevel identifier (`strengthlevel_slug`), required in the full prescription environments to map exercises to reference strength curves (see Section 4.2.3)

The union over all exercises defines the global muscle set $\mathcal{M}_{\text{all}}$ of size $|\mathcal{M}_{\text{all}}|$, used for muscle-level features.

For environments that require load estimation, the catalog is restricted to exercises with valid `strengthlevel_slug` mappings. This ensures prescription action and reward components relying on this data are well-defined. After filtering on valid `strengthlevel_slug` values, the final action space contains $|\mathcal{E}| = 217$ exercises. The same filtered catalog is also used in the exercise-only environments to keep the action space consistent across all settings.

4.2.2. User representation

At the start of each episode, a user u is sampled from a synthetic user pool. In the final implementation, each base user profile contains:

- age,
- sex,
- height,
- weight,
- goal,
- experience level,
- training frequency.

These features were chosen because they are both practically observable and directly relevant to recommendation quality, prescription realism, or user-simulator dynamics.

Age is included because the simulator targets adult gym users while still allowing variation across younger and older adults. In the Netherlands, fitness participation is mainly adult and remains substantial across both 18–64 and 65+ groups [70, 68]. Sex is included because both men and women participate in sport at broadly similar rates in the Netherlands [68], and because sex-specific strength references are used in the full-prescription environments.

Height and weight are included to create plausible body-size profiles and because weight directly affects the prescription-related strength estimates. Dutch anthropometric data show clear sex differences in adult height [59], while Dutch public-health reporting and survey data show that adults are spread across multiple body mass index (BMI) categories rather than one narrow weight class [58, 59]. For that reason, height is sampled from broad ranges and weight is derived from height together with a clipped BMI value rather than sampled independently.

Goal and experience are modeled categorically because resistance-training prescription depends on both training objective and training status. The progression models created by the American College of Sports Medicine distinguish strength, hypertrophy, and muscular endurance as different training targets and emphasize that appropriate prescription depends on experience level [4]. Training frequency is included because weekly session frequency affects progression and recovery [4], and weekly sport participation is common in the Dutch context [68]. Since the exact real-world prevalence of goals and experience levels is not central to the simulator, these categories are sampled uniformly to ensure that all major recommendation regimes are represented during training.

For dynamic user pools, users additionally maintain a small long-term state that evolves between sessions. In the full-prescription environments this includes in particular a bounded strength multiplier, which is used for load mapping and prescription scoring. In the skip environments, users additionally contain hidden skip-related parameters and online skip-state estimates, described in Section 4.3.

4.2.3. Prescription discretization

In the full-prescription setting, an action specifies both the exercise and its prescription:

$$a_t = \left(e_t, b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right), \tag{4.5}$$

where e_t is the exercise index and $\left(b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right)$ are discrete bin indices for sets, reps, and load.

- Sets bins $\mathcal{B}_{\text{set}}$: small set of values (e.g., 2–6).
- Reps bins $\mathcal{B}_{\text{rep}}$: wider range (e.g., 2–20).
- Load bins $\mathcal{B}_{\text{load}}$: a fixed number of bins representing a fraction of a baseline one-repetition maximum (1RM).

Load mapping

A load bin b_t^{load} is mapped to a target fraction $f(b_t^{\text{load}})$ between `load_min_frac` and `load_max_frac` via linear interpolation. This fraction is applied to a baseline 1RM obtained from StrengthLevel using `strengthlevel_slug`, sex, and experience. If the user has a strength multiplier `strength_mult` (dynamic pools), the baseline 1RM is scaled by this factor before converting to a working weight. The resulting target 1RM is converted into an actual working weight for the chosen rep count using the inverted Epley relation [16, 44, 71]:

$$1\text{RM} \approx w(1 + \frac{\text{reps}}{30}) \quad \Rightarrow \quad w \approx \frac{1\text{RM}}{1 + \text{reps}/30} \tag{4.6}$$

Weights are rounded to the nearest 0.5kg.

4.2.4. Observation encoding

This section specifies the concrete encoding of the observation o_t introduced in Section 4.1.4. The agent receives a fixed-length observation vector combining user information, session context, and, where applicable, skip-related interaction signals.

In all environments, the observation contains:

1. normalized scalar features for age, body weight, training frequency, and within-session step progress,
2. one-hot encodings for goal and experience level,
3. a one-hot routine indicator for the sampled session routine,
4. an exercise-history mask of length N indicating which exercise indices have already been suggested in the current session.

The full-prescription environments additionally include a two-dimensional one-hot indicator showing whether the previous session had the same routine as the current session. This is used because the prescription reward includes a recovery-aware adjustment when the same routine repeats.

The skip environments extend the observation with skip-state variables that make within-session adaptation possible:

1. a normalized current workload proxy,
2. a per-muscle strain vector over $\mathcal{M}_{\text{all}}$,
3. a binary last-skip indicator,
4. an exponential moving average of recent skipping,
5. normalized online estimates of capacity and skip bias,
6. a per-muscle avoidance estimate vector.

Importantly, these are estimates and interaction summaries rather than the simulator's hidden true parameters. The observation is therefore informative enough for adaptation, but still does not expose the full hidden user state directly.

4.2.5. Session history representation

To support inter-session reward components and dynamic users, each environment maintains a lightweight session history for every user. At minimum, a history entry stores:

- routine label for the session.
- list of recommended(/completed) exercise IDs (and sets, reps, load).

4.3. User simulators

Because large-scale real user interaction data is not available for this task, we use synthetic users and a user simulator to generate training episodes. The simulator serves two purposes. First, it provides diverse user profiles so that policies can be trained across different user types. Second, in the skip environments, it generates implicit feedback signals that imitate realistic user behavior.

4.3.1. Static user pools

A static user pool is generated once at the start of training. Each episode then samples one user uniformly from this fixed pool. The pool is intended to represent a diverse but bounded synthetic population rather than a dataset fit to one specific real gym.

User attributes are sampled as follows. Age is drawn uniformly between 18 and 70. Sex is sampled randomly. Height is sampled from a sex-conditioned uniform range, after which weight is derived from height and a clipped BMI sample. Goal and experience are sampled categorically, and training frequency is sampled as a bounded integer. In the skip pools, users additionally receive latent skip-behavior parameters drawn from experience-conditioned distributions, so that beginners, intermediates, and advanced users differ in both profile variables and in simulated tolerance and skip behavior.

Static pools provide a controlled setting in which the user distribution does not change over time. This makes them useful for reproducing the HFRL-style setting and for isolating the effect of action-space complexity, skipping, and online personalization.

4.3.2. Dynamic user pools

To test whether learned policies remain effective when users change over time, we also use dynamic user pools. In these pools, each user keeps a small long-term state that evolves between sessions.

All dynamic pools update age through an internal day counter, maintain a bounded strength multiplier, and track completed-session counts for possible experience progression. Weight also changes gradually over time through a goal-dependent drift rule with BMI-based bounds. In the skip pools, the user's latent true capacity additionally evolves between sessions, so the tendency to skip is not fully stationary.

After each episode, the environment forms a short episode summary containing quantities such as completion ratio, skipped ratio, and a stimulus proxy. It also simulates a gap in days based on training frequency. These two inputs are then used to update the user state. High completion and useful training stimulus push the user toward gradual improvement, while inactivity or repeated skipping can lead to mild detraining. Experience level can increase after enough completed sessions together with sufficient long-term progress.

The purpose of the dynamic pools is moderate and structured user drift. The same user becomes slightly different over time, allowing us to test whether policies remain stable when the training distribution is no longer perfectly stationary.

4.3.3. Interactions: no-skip vs skip

We consider two interaction settings. In the no-skip environments, every suggested exercise is treated as completed. In the skip environments, the simulator decides whether the user skips a recommendation based on a skip-probability model. This makes it possible to model implicit feedback and adherence within the session.

4.3.4. Skip behavior model: hidden truth vs online estimate

Skip environments use two skip-probability functions:

- $p_{\text{true}}(\text{skip} \mid \cdot)$: the simulator's true skip probability, computed from hidden user variables and used to sample whether the user actually skips,
- $p_{\text{hat}}(\text{skip} \mid \cdot)$: the environment's online predicted skip probability, computed from the current estimated user state and exposed indirectly through the observation.

This reflects the practical setting in which a recommender does not know the user's true current capacity, but must infer it from interaction.

The main driver of skipping is the relation between accumulated within-session workload W_t and a user-specific capacity parameter. In the exercise-only skip environment, workload increases through exercise-level muscle demand and recent repetition of muscle groups. In the full-prescription skip environment, workload additionally depends on the estimated demand of the chosen prescription, including sets, reps, and load.

Besides workload and capacity, both skip probabilities also depend on:

- recent repetition or strain on muscles already trained in the session,
- per-muscle recency through `days_since_last_group`,
- late-session position through the step index,
- a routine-mismatch boost when the suggested exercise does not match the current session routine,
- a small base skip rate.

Both p_{true} and p_{hat} are implemented as logistic functions over these features. The key difference is that p_{true} uses the user's latent true parameters, while p_{hat} uses online estimates and can additionally include a learned skip-bias term and per-muscle avoidance estimates.

4.3.5. Online personalization (overview)

When a user skips, the environment produces a binary feedback signal $y_t \in \{0, 1\}$ indicating whether the suggested action was skipped. In the skip environments, this feedback is used to update an online user model during the episode, and the updated estimates are included in the next observation o_{t+1}.

In the final implementation, online personalization updates three estimated quantities:

- a skip-bias term b_{hat},
- a capacity estimate $\hat{C}$,
- per-muscle avoidance estimates $a_{\text{hat}}(m)$.

4.4. Online personalization from skip-only feedback

This section explains how the online user model is updated using skip-only feedback. The main idea is that the environment maintains an estimated view of the user's tolerance and avoidance behavior, and updates this estimate whenever a skip occurs.

4.4.1. True skip vs predicted skip

For a suggested action at step t, the simulator computes:

- a true skip probability $p_{\text{true},t}$
- a predicted skip probability $p_{\text{hat},t}$

The environment samples the actual outcome:

$$y_t = \begin{cases} 1 & \text{if the user skips,} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

4.4.2. Online update rule

After observing y_t, we compute the prediction error:

$$\text{err}_t = y_t - p_{\text{hat},t} \tag{4.8}$$

We then update three quantities:

Skip bias

$$b_{\text{hat}} \leftarrow \text{clip}(b_{\text{hat}} + \eta_b \text{err}_t, b_{\text{min}}, b_{\text{max}}) \tag{4.9}$$

This one captures if the user currently skips more or less than expected, independent of context.

Capacity estimation

$$\hat{C} \leftarrow \text{clip}(\hat{C} - \eta_C \, \text{err}_t, C_{\min}, C_{\max}) \tag{4.10}$$

If the user skips more than predicted, the update decreases $\hat{C}$, which increases future predicted skipping when workload is high.

Per-muscle avoidance

For each muscle m involved in the suggested exercise:

$$a_{\text{hat}}(m) \leftarrow \text{clip}(a_{\text{hat}}(m) + \eta_m \frac{\text{err}_t}{|\mathcal{M}(e_t)|}, a_{\text{min}}, a_{\text{max}}) \tag{4.11}$$

This pushes the model toward higher predicted skipping when recommending muscle groups that appear to trigger skips.

Bias and capacity are scalar updates, muscle avoidance is a small distributed update over the muscles hit by the exercise.

How personalization affects the RL agent

The updated estimates are included in the next observation o_{t+1}. This creates a closed loop; the policy can learn to recommend actions that both score well on session-quality reward and remain within the user's own personal capacity to reduce skipping.

In the HFRL framework, they also use skips in the user simulator, but only implicitly do these skips affect the rewards. No personal estimates are used.

4.5. Environments

Based on the action-space choices and interaction settings introduced earlier, we define four environments. Together, these environments make it possible to separate the effects of action-space complexity, user interaction, and online personalization.

A useful way to view the design is that every environment specifies:

- Action space
- Observation encoding
- No-skip vs skip
- Reward components

4.5.1. Overview of the four environments

The four environments differ along two dimensions: whether the agent recommends only exercises or full prescriptions, and whether skipping with online personalization is enabled.

Let $\hat{S} = (\hat{a}_1, \ldots, \hat{a}_T)$ denote the suggested sequence and S the completed sequence. In no-skip environments, $S = \hat{S}$. In skip environments, $|S| \leq T$.

Environment	Action space	Skips	Online pers.	Reward basis
`ExerciseOnlyNoSkip`	Discrete(N)	No	No	Suggested session
`ExerciseOnlySkip`	Discrete(N)	Yes	Yes	Completed session, scaled by completion
`FullPrescriptionNoSkip`	MultiDiscrete($N, \|\mathcal{B}_{\text{set}}\|, \|\mathcal{B}_{\text{rep}}\|, \|\mathcal{B}_{\text{load}}\|$)	No	No	Suggested session
`FullPrescriptionSkip`	MultiDiscrete($N, \|\mathcal{B}_{\text{set}}\|, \|\mathcal{B}_{\text{rep}}\|, \|\mathcal{B}_{\text{load}}\|$)	Yes	Yes	Completed session, scaled by completion

Table 4.1: Overview of the four environments

All four environments sample a session routine (push/pull/legs) at reset time and track within-episode history with an exercise mask. Each environment can be paired with either a static user pool or a dynamic user pool. When dynamic user pools are used, the environment updates the user pool after each episode using episode summary statistics.

4.5.2. `ExerciseOnlyNoSkip` (HFRL framework replication with our gym-domain data)

The goal of this environment is to recommend a sequence of gym exercises that form a coherent session for a specific user.

- Action: $a_t \in \{0, ..., N-1\}$ selects an exercise index.
- Observation: User profile, one-hot session routine, step progress, plus an exercise history mask indicating which exercises have already been chosen.
- Dynamics: Every recommended exercise is treated as completed. The environment simply appends the chosen exercise to the session list.
- Reward: Terminal reward evaluates the full exercise sequence after T steps using session-quality components (see Section 4.6). Intermediate rewards are zero.

This environment is intentionally close to the HFRL-style of session-quality optimization and is used to reproduce baseline behavior in a gym catalog.

4.5.3. `ExerciseOnlySkip`: exercise-only with skipping + online personalization

The goal of this environment is to introduce realistic execution behavior. The user may skip exercises, and the environment maintains an online estimate of skip-related user parameters updated from skip-only feedback.

- Action: Same as `ExerciseOnlyNoSkip`
- Observation: In addition to the `ExerciseOnlyNoSkip` features, `ExerciseOnlySkip` exposes signals that make skipping learnable and personalizable, typically including:
  - a normalized current workload proxy,
  - per-muscle strain vector,
  - last-skip indicator and a skip EMA,
  - normalized online estimates,
  - a per-muscle avoidance estimate vector.
- Dynamics:
  1. The agent proposes an exercise.
  2. The simulator samples `did_skip` using p_{true}

3. If not skipped, the exercise is added to S and increases workload/strain/ If skipped, only the skip counter updates.
4. The environment updates the online skip model parameters using the feedback, producing the updated online estimates.

- Reward: Terminal reward evaluates completed behavior and is computed from the reward components and then scaled by the completion ratio. This ensures the agent is rewarded for recommending sessions that users actually perform, not just sessions that look good on paper.

4.5.4. `FullPrescriptionNoSkip`: full prescription planning, no skipping

The goal of this environment is to extend from selecting exercises to recommending full gym prescription (exercise + sets + reps + load), introducing rewards that only make sense in a gym context.

- Action: $a_t = \left(e_t, b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right)$ with e_t an exercise index and $\left(b_t^{\text{set}}, b_t^{\text{rep}}, b_t^{\text{load}}\right)$ bin indices for sets, reps, and load.
- Observation: Similar user profile and routine features as before, plus step progress and an exercise mask. FullPrescriptionNoSkip also includes a small indicator for whether the previous session had the same routine (used for recovery-aware prescription scoring).
- Dynamics: No skipping, so every prescription is completed and appended to the session plan.
- Reward: Terminal reward is computed from the full prescription sequence and combines:
  - sequence quality,
  - prescription quality based on how reasonable sets/reps/load are,
  - persistence comparing to the last session with the same routine.

4.5.5. `FullPrescriptionSkip`: full prescription + skipping + online personalization

The goal of this environment is to combine full prescription planning with realistic user interaction. The user may skip overly demanding prescriptions, and the system must learn under implicit feedback.

- Action: Same as FullPrescriptionNoSkip.
- Observation: Includes the full FullPrescriptionNoSkip observation together with ExerciseOnlySkip-style interaction signals.
- Dynamics:
  1. The agent proposes a prescription.
  2. The environment estimates the demand of this prescription.
  3. The simulator samples skipping using p_{true}.
  4. Online parameters are updated from the skip feedback
  5. Only completed prescriptions contribute to the performed session S and to workload accumulation.
- Reward: Terminal reward uses the same `FullPrescriptionNoSkip` reward components but is applied to completed prescriptions. To reflect adherence, the final session reward is scaled by completion, ensuring that great but unperformed plans are discouraged.

`FullPrescriptionSkip` is the most complete setting in this thesis. The agent must plan prescriptions under uncertainty about user capacity and learn to trade off training quality against adherence.

4.6. Reward design

The reward function is designed to capture the main qualities of a good gym session. Depending on the environment, it combines sequence-level qualities such as diversity, routine alignment, and fatigue management with adherence terms and, in the full-prescription setting, prescription quality.

All environments use a terminal, session-level reward that comprises several reward components $R_k \in [0, 1]$, which are then combined into a single scalar episode reward returned at the end of the session.

In Table 4.2 all reward components are explained in short.

Component	Meaning	EONS	EOS	FPNS	FPS
Uniqueness R_{unique}	Rewards sessions without repeated exercises.	Yes	Yes	Yes	Yes
Intra-session diversity R_{intra}	Rewards variation in muscle usage within the same session.	Yes	Yes	Yes	Yes
Inter-session diversity R_{inter}	Rewards variation compared with the user's previous session.	Yes	Yes	Yes	Yes
Routine match R_{routine}	Rewards alignment with the sampled session routine (push/pull/legs).	Yes	Yes	Yes	Yes
Fatigue R_{fatigue}	Penalizes repeated loading of the same muscles in close succession.	Yes	Yes	Yes	Yes
Completion scaling	Scales the final reward by the fraction of recommended items that are actually completed.	No	Yes	No	Yes
Prescription quality R_{rx}	Rewards plausible and user-appropriate sets, reps, and load.	No	No	Yes	Yes
Persistence R_{persist}	Rewards consistency with the user's previous session of the same routine.	No	No	Yes	Yes

Table 4.2: Reward components used in each environment. EONS = ExerciseOnlyNoSkip, EOS = ExerciseOnlySkip, FPNS = FullPrescriptionNoSkip, and FPS = FullPrescriptionSkip.

4.6.1. Notation and shared structure

A session has fixed length T. Let $\hat{S} = (\hat{a}_1, \ldots, \hat{a}_T)$ be the sequence of suggested actions and S the sequence of completed actions. In no-skip environments, $S = \hat{S}$. In skip environments, $|S| \leq T$.

Each completed action corresponds to an exercise index $e_i \in \{0, ..., N-1\}$. We use:

- $\mathcal{M}(e)$: the set of muscles associated with exercise e (parsed from both target and secondary muscle fields),
- routine(e): the routine label
- $S = (e_1, \ldots, e_{|S|})$: the completed exercise sequence used for reward computation (exercise-only environments).

For a set of components $\{R_k\}_{k \in \mathcal{K}}$, each component has:

- a penalty threshold $\tau_k \in [0, 1]$,
- a weight $w_k \geq 0$ with $\sum_{k \in \mathcal{K}} w_k = 1$.

We implement a simple penalty shaping:

$$\tilde{R}_k = \begin{cases} -0.5 & \text{if } R_k < \tau_k, \\ R_k & \text{otherwise,} \end{cases} \qquad R_{\text{total}} = \text{clip}\Big(\sum_{k \in \mathcal{K}} w_k \, \tilde{R}_k, \, -1, \, 1 \Big). \tag{4.12}$$

This hard penalty makes the training signal more separable. Sessions that violate important criteria become clearly worse than session that satisfy all criteria.

The HFRL framework uses per-component thresholds for penalization and additional boosting and notes that if any component is penalized the episode reward becomes negative. Our implementation keeps the same penalization spirit by using per-component thresholds and a fixed penalty value of -0.5 when a component falls below its threshold, but we do not implement an additional boosting threshold.

4.6.2. Thresholds and weights used in the final implementation

The previous subsection introduced the general aggregation rule. In the final implementation, the exact thresholds and component weights are fixed per action-space family. Because the skip environments

use completion scaling rather than an explicit completion component, `ExerciseOnlySkip` uses the same thresholds and component weights as `ExerciseOnlyNoSkip`, and `FullPrescriptionSkip` uses the same thresholds and component weights as `FullPrescriptionNoSkip`. The skip variants therefore differ through reward basis and completion scaling, not through different per-component weights.

Component	Exercise-only		Full prescription	
	Threshold	Weight	Threshold	Weight
R_{unique}	0.60	0.20	0.80	0.09
R_{intra}	0.60	0.20	0.60	0.05
R_{inter}	0.00	0.20	0.50	0.08
R_{routine}	0.60	0.20	0.50	0.22
R_{fatigue}	0.10	0.20	0.30	0.14
R_{rx}	–	–	0.50	0.32
R_{persist}	–	–	0.50	0.10

Table 4.3: Per-component thresholds and weights used in the final implementation. Because skip environments use completion scaling, they share the same thresholds and component weights as their corresponding no-skip variants.

4.6.3. Uniqueness reward R_{unique}

To explicitly discourage repeated recommendations within a session, we compute the set of distinct exercise indices in the session and divide by the session length:

$$R_{\text{unique}}(S) = \frac{|\{e \mid e \in S\}|}{|S|}. \tag{4.13}$$

This yields 1 when all completed exercises are unique and decreases when repeats occur.

The HFRL framework does not include a standalone uniqueness term. They do include a penalization factor in the intra-session diversity reward component, but we add the uniqueness reward as a more direct session-level repetition signal.

4.6.4. Intra-session diversity R_{intra}

This component rewards diversity of muscle usage within the same session. We define muscle-set similarity between two exercises using Jaccard similarity:

$$\text{sim}(e_i, e_j) = \frac{|\mathcal{M}(e_i) \cap \mathcal{M}(e_j)|}{|\mathcal{M}(e_i) \cup \mathcal{M}(e_j)|}. \tag{4.14}$$

To emphasize local redundancy more than distant redundancy, we apply a distance-based decay. Let $0 < w_{\text{decay}} < 1$. For positions $i < j$, define $\Delta = j - i$ and weight $w_{\text{decay}}^{\Delta-1}$. Then we compute a weighted mean similarity and invert it:

$$R_{\text{intra}}(S) = 1 - \frac{\sum_{i<j} w_{\text{decay}}^{(j-i)-1}\, \text{sim}(e_i, e_j)}{\sum_{i<j} w_{\text{decay}}^{(j-i)-1} + p}. \tag{4.15}$$

Here p is consecutive-repeat penalty:

$$p = \begin{cases} -1 & \text{if } \exists i \text{ such that } e_i = e_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

In our implementation we:

- Parse each exercise into a muscle set $\mathcal{M}(e)$ (target + secondary)
- For each pair (i, j), compute Jaccard similarity and accumulate with decay weight
- If any consecutive pair repeats the exact same exercise, set $p = -1$.
- Compute the diversity score $1 - \frac{\text{weighted similarity}}{\text{weighted sum} + p}$.

This is the same structure as HFRL's intra-session diversity. Our main change is using Jaccard similarity on the union of target+secondary muscles, whereas the paper describes similarity as "mean similarity of the muscles they train".

4.6.5. Inter-session diversity R_{inter}

Inter-session diversity encourages variation across sessions by comparing the muscles trained in the current session to those trained in the user's immediately preceding session (the last overall session, not necessarily the last same-routine session). Let $M(S)$ denote the set of muscles trained in a session S (constructed from target and secondary muscles). We compute a similarity between the two sessions using Jaccard similarity and define inter-session diversity as its complement:

$$\text{sim}_{\text{set}}(S, S^-) = \frac{|M(S) \cap M(S^-)|}{|M(S) \cup M(S^-)|}, \qquad R_{\text{inter}}(S) = 1 - \text{sim}_{\text{set}}(S, S^-). \tag{4.17}$$

If the user has no previous session history, inter-session diversity defaults to a configured value (in our experiments we treat it as maximally diverse).

4.6.6. Routine-match reward R_{routine}

Gym programs are often structured by a routine/day split. In our data we only use three routines: push, pull and legs. We sample a target routine r^* per session and reward selecting exercises from that routine:

$$R_{\text{routine}}(S) = \frac{1}{|S|} \sum_{i=1}^{|S|} \mathbf{1}\big(\text{routine}(e_i) = r^\star\big). \tag{4.18}$$

This returns 1 if all completed exercises match the intended routine and decreases linearly with mismatches.

The HFRL framework uses focus muscle matching R_{musc}, where users specify primary and secondary muscle sets and the reward checks whether each exercise's muscle set lies within those targets. As gym training commonly follows routine days, we replace these focus muscles with routines. Conceptually, both terms enforce that the session aligns with an intended target structure, but with routines we can better keep persistent training structures over multiple sessions by keeping the choice smaller and less varied.

4.6.7. Fatigue reward R_{fatigue}

This component penalizes repeatedly training the same muscle groups in close succession. For each muscle m, we collect the list of positions where it appears in the session. In contrast to a pure set-based formulation, we weight target muscles higher than secondary muscles: target hits use weight 1.0 and secondary hits use a smaller weight (0.35 in our implementation). For each pair of occurrences (i, j) of the same muscle, we accumulate a decay-weighted contribution:

$$F(m) = \sum_{i<j} w_i w_j w_{\text{decay}}^{(j-i)-1}, \qquad F_{\max} = \max_m F(m). \tag{4.19}$$

We convert this to a reward in $[0, 1]$ using an exponential transform:

$$R_{\text{fatigue}}(S) = \exp\left(-\frac{F_{\max}}{\tau_{\text{fat}}}\right), \tag{4.20}$$

where τ_{fat} controls how quickly the reward decays as repetition increases.

4.6.8. Completion/skipping terms

Skip environments measure adherence via the completion ratio, defined as the fraction of suggested steps that were completed:

$$c = \frac{|S|}{T}, \tag{4.21}$$

where S is the completed sequence and T is the fixed session length. By default, we incorporate adherence using reward scaling: we compute the base terminal reward from plan-quality components (and prescription components in full-prescription environments) and then multiply by c:

$$R_{\text{final}} = c \cdot R_{\text{total}}. \tag{4.22}$$

This discourages plans that look good under the session-quality metrics but are not executed. As an alternative configuration, completion can also be included as an explicit weighted reward component rather than scaling, but scaling is the default used in our experiments.

4.6.9. Prescription quality reward R_{rx}

The full prescription environments introduce a prescription-quality component R_{rx} that evaluates whether the recommended sets, reps, and load are plausible and personalized for the user.

The key idea is: a good prescription should resemble common sets and reps for the exercise and choose a load that matches the user's expected strength. We also need to stay conservative when the session repeats the same routine as the last session.

Sets+reps plausibility term q_{sr}

For each exercise, the StrengthLevel dataset provides a set of popular workouts (typical pairs of sets and reps for a specific exercise). We denote $\mathcal{W}_{\text{pop}}(e)$ as the set of popular $(\text{sets}, \text{reps})$ prescriptions for exercise e from StrengthLevel. Given $(\text{sets}_i, \text{reps}_i)$, we score how close it is to these popular prescriptions using a Gaussian kernel, and take the best match:

$$q_{\text{sr}} = \max_{(\bar{s},\bar{r})\in\mathcal{W}_{\text{pop}}(e_i)} \exp\Big(- \frac{(\text{sets}_i - \bar{s})^2}{2\sigma^2_{\text{sets}}} - \frac{(\text{reps}_i - \bar{r})^2}{2\sigma^2_{\text{reps}}} \Big). \tag{4.23}$$

If no popular workouts exist for an exercise, the implementation falls back to a constant neutral score of 0.5.

Load appropriateness term q_{load}

The load term compares the chosen load to user-specific target intensity.

We first convert the chosen load to an estimated 1RM. The agent chooses load_i and reps_i. We convert that to an implied 1RM via Epley:

$$1\text{RM}_{\text{chosen}} \approx \text{load}_i \Big(1 + \frac{\text{reps}_i}{30}\Big). \tag{4.24}$$

This gives a load-independent representation of intensity.

Next, we compute the target 1RM for this user and exercise. Using StrengthLevel, we compute an expected 1RM for the user, $1\text{RM}_{\text{expected}}$, based on the exercise + user attributes (sex, age, bodyweight, experience). We then compute a baseline 1RM, 1RM_{base}, based on the exercise, sex and experience. The user may also have a dynamic training multiplier `strength_mult`, and when it exists we scale the expected 1RM accordingly:

$$1\text{RM}_{\text{expected}} \leftarrow 1\text{RM}_{\text{expected}} \cdot \text{strength\_mult}. \tag{4.25}$$

We then compare on a normalized scale. Rather than comparing kilograms directly (which differs a lot across exercises), we compare fractions of the baseline 1RM:

$$\text{chosen\_frac} = \frac{1\text{RM}_{\text{chosen}}}{1\text{RM}_{\text{base}}}, \qquad \text{target\_frac} = \frac{1\text{RM}_{\text{expected}}}{1\text{RM}_{\text{base}}}. \tag{4.26}$$

As mentioned previously, if the previous session's routine matches the current routine, the code reduces the target intensity to reflect incomplete recovery:

$$\text{target\_frac} \leftarrow \alpha \cdot \text{target\_frac}, \qquad \alpha \in (0, 1]. \tag{4.27}$$

As a last step for the load term we score using a log-ratio Gaussian. We score how close chosen and target fractions are using a Gaussian in log space:

$$\rho = \frac{\text{chosen\_frac}}{\text{target\_frac}}, \qquad q_{\text{load}} = \exp\Big(-\frac{\log^2(\rho)}{2\sigma^2_{\text{load}}}\Big). \tag{4.28}$$

Overshooting by a factor and undershooting by the same factor are penalized equally this way.

If any of the StrengthLevel values are missing or invalid, the implementation falls back to a conservative constant score to avoid noise.

Combine into a per-step score q_i

Each step's prescription score is a convex combination:

$$q_i = w_{\text{load}}\, q_{\text{load}} + w_{\text{sr}}\, q_{\text{sr}}, \qquad w_{\text{load}} + w_{\text{sr}} = 1. \tag{4.29}$$

We default to $w_{\text{load}} = 0.60$ and $w_{\text{sr}} = 0.40$. This means load accuracy matters slightly more than matching common set/rep patterns.

Aggregate across the session with a weakest link penalty

A simple average could hide a single terrible prescription step inside an otherwise good session. To prevent this, we compute:

- The mean:
$$\bar{q} = \frac{1}{|S|}\sum_i q_i. \tag{4.30}$$
- and a soft-minimum:
$$\text{softmin}_\beta(\{q_i\}) = -\frac{1}{\beta}\log\Big(\frac{1}{|S|}\sum_i \exp(-\beta q_i)\Big). \tag{4.31}$$
with β controlling how close this behaves to a true minimum.

If softmin_β falls below a threshold (0.25), we subtract a penalty proportional to the shortfall (capped at 0.40). The final prescription reward is then:

$$R_{\text{rx}} = \text{clip}(\bar{q} - \text{penalty},\, 0,\, 1). \tag{4.32}$$

4.6.10. Persistence reward R_{persist}

Persistence rewards consistency across multiple session of the same routine, while still allowing improvement.

Let S be the current completed prescription list for the session and S^- the most recent previous session with the same routine. We compute two sub-scores:

1. Exercise persistence:
$$R_{\text{ex}} = \frac{|E(S) \cap E(S^-)|}{|E(S^-)|}, \tag{4.33}$$
where $E(\cdot)$ extracts the set of exercises used in a session.
2. Sets+reps persistence: for each shared exercise e, compare mean sets and reps between session using a Gaussian similarity:
$$R_{\text{sr}}(e) = \tfrac{1}{2}\exp\Big(-\frac{(\Delta\text{sets})^2}{2\sigma^2_{\text{sets}}}\Big) + \tfrac{1}{2}\exp\Big(-\frac{(\Delta\text{reps})^2}{2\sigma^2_{\text{reps}}}\Big). \tag{4.34}$$
and average over shared exercises.

Finally:

$$R_{\text{persist}} = w_{\text{ex}} R_{\text{ex}} + w_{\text{sr}} \Big(\frac{1}{|\text{shared}|} \sum_{e \in \text{shared}} R_{\text{sr}}(e) \Big), \qquad w_{\text{ex}} + w_{\text{sr}} = 1. \tag{4.35}$$

If there is no same-routine past session, we return a neutral default (0.5).

This reward seemingly contradicts the inter-session diversity, however having persistency in gym exercises for routines is common and is beneficial for muscle growth. Inter-session diversity still tries avoiding straining muscles between the current and last session (as secondary muscles might appear in multiple routines), whereas this persistence reward focuses on the current session in comparison with the last session of the same routine for this user.

4.7. Simulation process

This section describes how one simulated episode unfolds in the four environments. In all cases, an episode corresponds to one workout session of fixed length T.

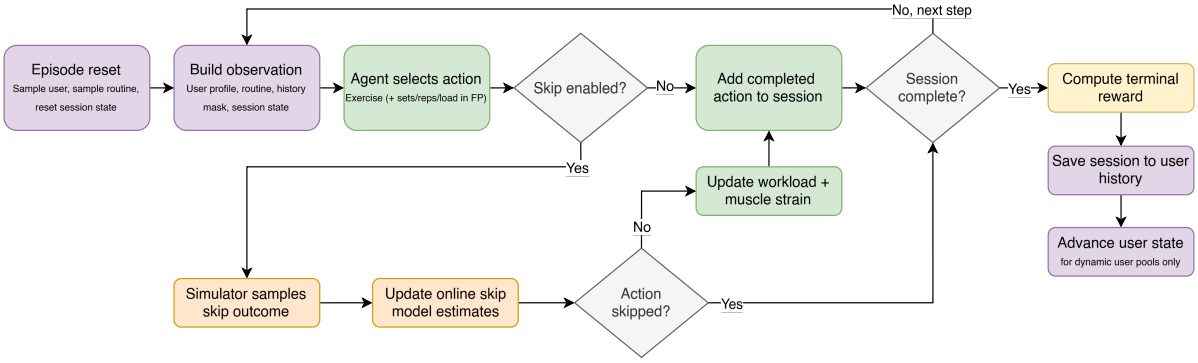


Figure 4.1: Simulation process for one episode. The agent constructs a session step by step. In skip-enabled environments, the simulator samples skip outcomes and the online skip model is updated from skip-only feedback. After the session is complete, a terminal reward is computed, the session is saved to user history, and dynamic user pools may advance the user state.

At the start of an episode, the environment samples one synthetic user from the selected user pool and initializes the session state. This includes the step counter, the within-session recommendation history, and a session routine drawn from the filtered exercise catalog. The initial observation then encodes the user profile, the current session progress, the selected routine, and the exercises already recommended. In the skip-enabled environments, the observation additionally includes workload, per-muscle strain, recent skip signals, and the online skip-related estimates used for personalization.

At each step, the agent proposes the next recommendation. In the exercise-only environments, the action is a single exercise index. In the full-prescription environments, the action consists of an exercise together with a sets bin, a reps bin, and a load bin. The chosen load bin is converted to a working weight in kilograms using the StrengthLevel lookup and the inverted Epley relation, so that the action represents a complete prescription rather than only an exercise choice.

The environments then differ in how user response is handled. In the no-skip environments, every proposed action is treated as completed. The environment therefore only appends the choice to the session and advances to the next step. In the skip-enabled environments, the proposed action is first passed through the user simulator. Based on the current workload, the user's latent capacity, repetition and recovery sensitivities, recent muscle strain, and the position within the session, the simulator computes a true skip probability. The environment also maintains an online estimate of this skip probability from the state. A skip outcome is then sampled from the true probability. If the action is completed, it is added to the performed session and the environment updates workload and muscle strain. If it is skipped, the performed session is left unchanged and only the skip-related counters are updated. In the full-prescription skip environment, the selected sets, repetitions, and load also affect the simulated skip probability through the estimated demand of the proposed prescription. After each skip decision, the

online personalization variables are updated from skip-only feedback, so that later recommendations within the same episode can adapt to the observed behavior.

Reward is terminal in all four environments. This means that the environment returns a non-zero reward only after the full session has been constructed. At the end of the episode, the reward function evaluates the resulting session using the components defined in Chapter 4.6. In the exercise-only environments, these components include uniqueness, intra-session diversity, inter-session diversity, routine matching, and fatigue-aware structure. In the full-prescription environments, prescription quality and persistence relative to the last same-routine session are added. In the skip environments, the final reward is based on the completed session and is scaled by the completion ratio, consistent with the reward configuration used in all experiments.

After the terminal reward has been computed, the completed session is written to the user history. This history is used in later episodes for components such as inter-session diversity and persistence. In the dynamic user pools, the user is then advanced to the next session by simulating a gap in days and applying long-term updates to age, weight, strength-related variables, and, in the skip-enabled environments, the latent capacity that influences future skipping. In the static pool no such update is applied, while the robust-dynamic and chaotic-dynamic pools apply stronger and deliberately less realistic changes for stress testing.

4.8. Illustrative recommendation examples

After defining the environments, rewards, and simulation process, it is useful to show what a recommendation produced by the trained PPO policies actually looks like. The examples below are included only to make the environments more concrete.

In the main text, we show one example from the simplest exercise-only setting and one from the most complete full-prescription setting. The remaining two examples are moved to Appendix C.1.

Example from `ExerciseOnlyNoSkip`

Tables 4.4–4.6 show a single recommendation generated by a trained PPO policy in the `ExerciseOnlyNoSkip` environment. Table 4.4 first shows the sampled user and the routine chosen by the environment. Based on this context, the model constructs the session shown in Table 4.5. Finally, Table 4.6 shows the resulting terminal reward and its component scores.

Feature	Value
User ID	346
Sex	male
Age	40
Height (cm)	162.82
Weight (kg)	52.47
Goal	endurance
Experience	intermediate
Training frequency	2
Sampled session routine	leg day

Table 4.4: Sampled user and session context for the illustrative `ExerciseOnlyNoSkip` recommendation.

This user is an intermediate trainee with an endurance goal, and the environment sampled a leg-day routine. The PPO policy therefore has to construct a session that fits this routine while still maintaining diversity and avoiding unnecessary repetition.

The resulting session is clearly structured around a leg-day theme as it mixes lower-body work with abdominal and cardio-related exercises. Because this environment does not model skipping and does not include prescription variables, the recommendation is only about which exercises should appear in the session.

The component scores show that this session scores perfectly on uniqueness, inter-session diversity, and routine match, while also achieving very high intra-session diversity and fatigue reward. This is

Step	Exercise	Target muscles	Secondary muscles	Equipment	Body part
1	barbell good morning	hamstrings	lower back	barbell	upper legs
2	sit-up v. 2	abs	hip flexors	body weight	waist
3	dumbbell standing calf raise	calves	ankles	dumbbell	lower legs
4	mountain climber	cardiovascular system	core, shoulders, triceps	body weight	cardio
5	side hip abduction	abductors	glutes, quadriceps	body weight	upper legs
6	lever leg extension	quads	hamstrings	leverage machine	upper legs
7	reverse crunch	abs	hip flexors	body weight	waist
8	lever seated calf raise	calves	soleus, ankle stabilizers	leverage machine	lower legs

Table 4.5: Illustrative recommendation produced by PPO in the `ExerciseOnlyNoSkip` environment.

Metric	Value
Total reward	0.986
R_{unique}	1.000
R_{intra}	0.992
R_{inter}	1.000
R_{routine}	1.000
R_{fatigue}	0.939

Table 4.6: Terminal reward and component scores for the illustrative `ExerciseOnlyNoSkip` recommendation.

useful to see before the reward definition, because it gives a concrete impression of what a high-scoring session in the exercise-only setting looks like.

Example from `FullPrescriptionSkip`

Tables 4.7–4.9 show a single recommendation generated by a trained PPO policy in the `FullPrescriptionSkip` environment. As before, we first show the sampled user and session context, then the actual recommendation, and finally the resulting reward scores.

Feature	Value
User ID	245
Sex	male
Age	50
Height (cm)	177.94
Weight (kg)	75.80
Goal	endurance
Experience	intermediate
Training frequency	2
Sampled session routine	push day
Days since last session	3
Strength multiplier	1.000
True capacity C_{true}	0.958
Estimated capacity $\hat{C}$	0.860
Estimated skip bias b_{hat}	0.000
Base skip rate	0.0017

Table 4.7: Sampled user and session context for the illustrative `FullPrescriptionSkip` recommendation.

In this case, the environment sampled a push-day session for an intermediate user. Because this is the full-prescription skip setting, the model does not only decide which exercises to suggest, but also how they should be prescribed through sets, repetitions, and load. At the same time, the environment keeps track of skip-related user variables.

Compared with the previous example, this recommendation is much richer. The session is still or-

Step	Exercise	Sets	Reps	Load (kg)	Target muscles	Secondary muscles	Equipment	Body part
1	cable standing rear delt row	3	6	33.5	delts	trapezius, rhomboids, biceps	cable	shoulders
2	cable reverse-grip pushdown	3	6	39.5	triceps	forearms	cable	upper arms
3	dumbbell fly	2	10	17.0	pectorals	shoulders	dumbbell	chest
4	burpee	2	9	21.5	cardiovascular system	quadriceps, hamstrings, calves, shoulders, chest	body weight	cardio
5	cable standing shoulder external rotation	2	9	13.5	delts	rotator cuff, trapezius	cable	shoulders
6	cable pushdown (rope attachment)	2	9	35.5	triceps	forearms	cable	upper arms
7	lever seated fly	2	10	63.0	pectorals	deltoids, trapezius	leverage machine	chest
8	dumbbell front raise	2	10	13.0	delts	biceps, trapezius	dumbbell	shoulders

Table 4.8: Illustrative recommendation produced by PPO in the `FullPrescriptionSkip` environment.

ganized around the sampled routine, but now every step also includes a dose recommendation. In this sampled episode, all suggested steps were completed, so the completed session is equal to the suggested session.

Metric	Value
Total reward	0.874
R_{unique}	1.000
R_{intra}	0.934
R_{inter}	1.000
R_{routine}	1.000
R_{fatigue}	0.677
R_{rx}	0.913
R_{persist}	0.803
Completion ratio	1.000

Table 4.9: Terminal reward and component scores for the illustrative `FullPrescriptionSkip` recommendation.

The reward scores show that the recommendation matches the intended routine perfectly and also scores highly on uniqueness, intra-session diversity, inter-session diversity, and prescription quality. The lower fatigue score is also informative, because it shows that even a strong overall recommendation can still involve trade-offs between different reward components.

4.9. Learning algorithms + baselines

To test whether reinforcement learning is actually beneficial in this setting, we compare PPO against several non-learning baselines. These baselines provide reference points ranging from uninformed recommendations to stronger hand-crafted or search-based strategies.

4.9.1. Baseline policies

We evaluate three baselines: a random policy, a greedy heuristic, and a PSO-based planner. Together, these baselines cover simple sampling, reward-aligned hand-crafted planning, and offline search over candidate sessions.

Random baseline

The random baseline samples an action uniformly from the environment's action space at every step:

- Exercise-only environments: uniform exercise index.
- Full prescription environments: uniform (`exercise, sets bin, reps bin, load bin`)

This baseline establishes a lower bound on performance and provides a sanity check that reward components and the environments behave as expected.

Greedy heuristic baseline

The greedy baseline is an interpretable heuristic that constructs a session sequentially. At each step it selects the action that maximizes a hand-designed score based on the same signals the RL agent observes.

For each candidate exercise e, the greedy policy computes a weighted score of:

1. Routine alignment
2. Uniqueness
3. Local diversity
4. Goal hint
5. Persistence
6. Strain avoidance
7. Noise for tie-breaks

The greedy baseline is a strong baseline as it encodes the most obvious structure a human designer would use, without actually learning any patterns.

When the environment is multi-discrete, the greedy policy chooses reps based on goal and sets and target load fraction based on experience and goal. The target fraction is reduced if the previous session is the same routine.

PSO planning baseline

The PSO baseline performs offline planning for each episode using an integer variant of Particle Swarm Optimization (PSO). Whereas the greedy policy optimizes one step at a time, PSO tries to plan the entire session at once.

For each episode, PSO does the following:

1. Define a plan representation.
    - Exercise-only: a vector of length T: $[e_1, ..., e_T]$
    - Full prescription: a vector of length T of tuples: $\left[(e_1, b_1^{\text{set}}, b_1^{\text{rep}}, b_1^{\text{load}}), \dots\right]$
2. Search over plans: PSO keeps a swarm of candidate plans. Over several iterations it updates these candidate plans to move towards better solutions.
3. Score a candidate plan: To evaluate one candidate plan, PSO:
    - creates a fresh environment with the same episode seed (so the same user and routine are used)
    - executes the plan step-by-step
    - reads the terminal reward,
    - use that as the plan's fitness
4. Return the best plan: After the set iterations, PSO keeps the best plan and then simply executes it during the episode.

5

Experimental Setup

This chapter describes how the experiments are carried out. It explains the data and environment configuration, the user pools, the training setup, the evaluation protocol, and the baselines used in the comparisons. The goal is to test whether RL is actually needed in our setting, what skipping and online personalization change, and how robust learned policies are under user feature changes.

5.1. Overview

We train PPO agents on the four environments introduced in Section 4.5. Each environment is combined with four user-pool regimes, resulting in $4 \times 4 = 16$ trained models:

- Environments: `ExerciseOnlyNoSkip`, `ExerciseOnlySkip`, `FullPrescriptionNoSkip`, `FullPrescriptionSkip`.
- User pools: `static`, `dynamic`, `robust_dynamic`, `chaotic_dynamic`.

The `static` and `dynamic` pools are used for the main comparisons, such as PPO versus baselines and no-skip versus skip. The `robust_dynamic` and `chaotic_dynamic` pools are included as stress tests to study robustness under stronger and intentionally unrealistic distribution shifts.

All experiments use the same session length $T = 8$ and the same reward configuration defined in Chapter 4.

5.2. Data and environment configuration

5.2.1. Why these data sources were used

A practical challenge in this thesis is that the kind of public data needed for a sequential gym recommender is largely missing. Ideally, one would want a dataset that combines a broad exercise catalog with session-level exercise order, prescription variables such as sets, repetitions, and load, user-specific context, and some realistic interaction signal such as skipping or adherence. In practice, public gym-related data is much more fragmented.

On the one hand, there are many content-focused exercise datasets that describe exercises well, including target muscles, equipment, and movement-related information. On the other hand, there are some workout logs, plan templates, and sensor datasets, but these usually miss one or more important parts of the recommendation problem. Executed gym logs are often too small, too narrow, single-user, or unstructured. Sensor and form datasets are useful for exercise recognition or coaching, but not for deciding what should be recommended next in a full session. As a result, no public source was found that cleanly supports sequential gym recommendation with both full prescription and user interaction.

For that reason, this thesis does not use a logged-data recommendation setup. Instead, it combines structured exercise content data with synthetic users and a simulator. The exercise catalog defines the available action space and the exercise metadata used in the reward and observation design.

StrengthLevel-derived data is used in the full-prescription environments to map exercises to plausible load references and to score prescription quality. User behavior and adherence are then modeled through the simulator, which makes controlled training and evaluation possible even without real session-level interaction logs.

A fuller overview of the data search that motivated this design choice is provided in Appendix D. The appendix is included for completeness, but the main practical consequence for the experiments is simple: the available public data was sufficient for defining the exercise space and prescription references, but not for directly learning a personalized sequential gym recommender from real interaction logs.

All environments use the same exercise spreadsheet derived from the ExerciseDB V1 dataset [18]. The full-prescription environments additionally use a StrengthLevel JSON file for load mapping and prescription scoring, as described in Section 4.2.3 [71].

5.2.2. Catalog filtering

To keep the action space consistent across environments, the exercise catalog is restricted to exercises with a valid `strengthlevel_slug`.

As a result, all environments operate on a single consistent exercise index space after filtering.

5.2.3. Prescription bins

For the full-prescription environments, sets, reps, and load are discretized into fixed bins that match the implementation used during training and evaluation:

- Sets bins: $(2, 3, 4, 5, 6)$
- Reps bins: $(2, 3, \ldots, 20)$
- Load bins: 21 bins over $[0.20, 1.20]$ of a baseline 1RM

When the last session routine matches the current routine, the recovery factor $\alpha = 0.70$ is applied as described in Section 4.6.

5.3. User pools

Each environment is paired with one of four user-pool regimes. In all experiments, the pool size is fixed to 500 synthetic users. The regimes are the following:

5.3.1. Static

The static pool is sampled once at initialization and does not change over episodes. This gives a stationary training distribution.

5.3.2. Dynamic

In the dynamic pool, users evolve gradually between sessions through two mechanisms:

- a gap model based on training frequency (`simulate_gap_days`),
- long-term state updates (`advance_long_term`).

These updates affect age, weight, strength-related state, and experience progression in all dynamic pools. In the skip pools, they also affect the latent true capacity that drives skipping. The updates are bounded and gradual, so the resulting drift is moderate rather than abrupt. The dynamic regime is therefore meant to represent realistic non-stationarity: users change over time, but not so strongly that their behavior becomes completely unrelated from one session to the next.

5.3.3. Robust dynamic (stress test)

The robust dynamic pool is a stress-test setting in which user variables change strongly in one direction after each episode. The goal is to test whether learned policies remain stable under larger structured shifts, not realism.

5.3.4. Chaotic dynamic (stress test)

The chaotic dynamic pool is a stronger stress test in which user variables can move up or down randomly between sessions. This creates a deliberately unstable setting for testing the limits of the learned policies.

5.4. Reward configuration used in all experiments

We keep the reward design fixed across training and evaluation (Section 4.6). Concretely, the experiments use a single global reward configuration:

- Completion handling in skip environments: scaling

$$R_{\text{final}} = c \cdot R_{\text{total}}, \qquad c = \frac{|S|}{T}.$$

- Plan basis in skip environments: completed plan (reward is computed on completed items).
- Inter-session diversity: muscle-based set Jaccard between current session and the last session (defaults to 1.0 if no history exists).
- Threshold shaping: if a component falls below its threshold, it is replaced by a fixed penalty value of -0.5 before weighting.
- Final reward clipping: $[-1, 1]$.

This ensures that differences in performance are caused by the environment/user dynamics and the learning algorithm, not by changing reward definitions.

The exact per-component thresholds and weights are those defined in Table 4.3 and remain fixed across all experiments.

5.5. RL training setup

All RL agents are trained using PPO with a shared configuration across all 16 runs:

- Policy: `MlpPolicy` with a 2-layer MLP, hidden sizes $[256, 256]$.
- Learning rate: $3 \cdot 10^{-4}$
- Rollout length: $n\_steps = 2048$
- Batch size: 256
- PPO epochs per update: 10
- Discount factor: $\gamma = 0.99$
- Entropy coefficient: 0.01
- Clip range: 0.2

Training uses a custom callback that logs the episode return and the terminal reward components (from `info["components"]`) at the end of each episode. Each model is trained for a fixed number of timesteps (typically 600000 for exercise-only environments to 1000000 for full prescription environments in our runs), and all models are saved after training for evaluation.

5.6. Evaluation protocol

We evaluate policies by rolling out complete episodes and recording both the overall return and the individual reward components. In the skip environments, we also record completion ratio and skipped ratio.

5.6.1. Fresh vs continual evaluation

We use two evaluation modes:

- Fresh evaluation: each episode is run in a freshly created environment instance (pool + env recreated), seeded per episode. This measures average performance under the pool distribution, but does not keep track of evolving users.
- Continual evaluation: a single environment instance is reused across episodes, so user history and their updates accumulate naturally. This is especially relevant for the dynamic/robust/chaotic pools.

Unless stated otherwise, we report the main results using continual evaluation to ensure an evaluation that mimics real-life scenarios.

5.6.2. Uncertainty and confidence intervals

To avoid over-interpreting small differences, mean returns are reported together with 95% bootstrap confidence intervals. The same procedure is used for skip ratio where relevant.

5.7. Baselines

To test whether RL is necessary, we compare PPO against three baselines. Baselines are evaluated on the same environments and reward configuration, and we focus this comparison on the `static` and `dynamic` pools (the realistic regimes). Robust/chaotic pools are used mainly for robustness analysis rather than baseline ranking. As mentioned in Section 4.9, we evaluate on:

- a random baseline,
- a greedy heuristic,
- and a PSO planner.

5.8. Planned comparisons and reporting structure

The results chapter is organized around four main comparisons. Together, these comparisons answer the sub-research questions and separate the effects of action-space complexity, user drift, skipping, and robustness.

Static vs dynamic training

We compare models trained on static pools versus dynamic pools. This tests whether PPO learns policies that remain effective when users gradually evolve over time.

No-skip vs skip environments

We compare `NoSkip` and `Skip` environments within each action-space family. This isolates what skipping and online personalization change in terms of reward components and learning curves.

RL vs baselines

For each environment in the realistic pool regimes, PPO is compared with random, greedy, and PSO. This is the main comparison for determining whether the added complexity of the planning problem justifies reinforcement learning.

Robustness under extreme drift

We evaluate policies trained on `robust_dynamic` and `chaotic_dynamic` pools to examine how performance changes under stronger and less realistic user drift. Again, realism is not desired here.

5.9. Implementation notes (reproducibility)

All training and evaluation code is implemented in Python using Gymnasium environments and Stable-Baselines3 PPO.

6

Results

6.1. Exercise-only setting: baseline comparison and replication

Across the four exercise-only comparisons (`ExerciseOnlyNoSkip` and `ExerciseOnlySkip`, each evaluated on both static and dynamic pools), the greedy baseline performs best and the random baseline performs worst, whereas PPO and PSO form the middle group. PPO is generally competitive with PSO, but the difference between them is small, which shows that reinforcement learning is able to learn useful structure in the recommendation task, but it does not clearly outperform the strongest non-learning baselines.

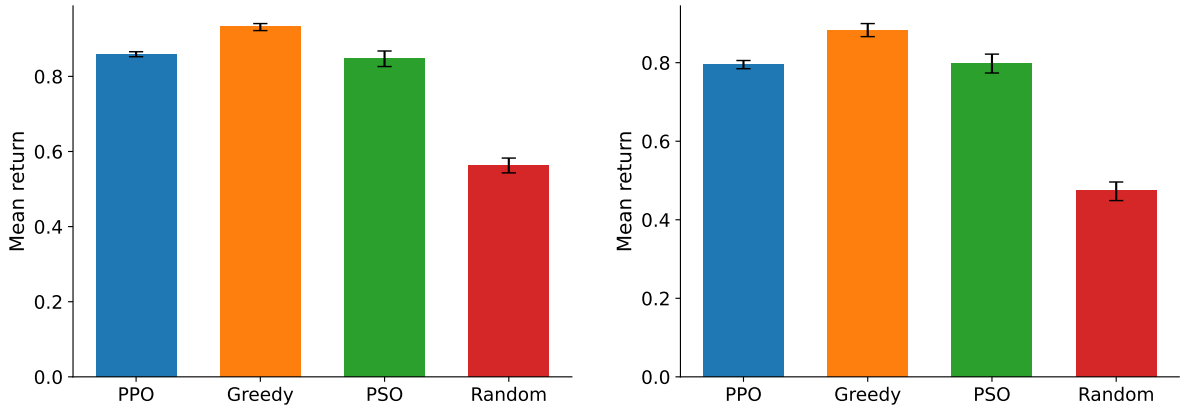
This partly reproduces the general pattern of the HFRL framework in a gym-domain setting. The `ExerciseOnlyNoSkip` environment is intentionally close to that earlier setup, but it is still adapted to the gym domain, with a different exercise catalog and a routine-based target term. The main result is therefore that the same broad trend appears in our setting: RL performs better than random recommendation and remains competitive with PSO, but a strong greedy baseline still performs best.

Environment	Pool	PPO	Greedy	PSO	Random
ExerciseOnlyNoSkip	Static	0.859	**0.932**	0.849	0.563
ExerciseOnlyNoSkip	Dynamic	0.863	**0.932**	0.849	0.563
ExerciseOnlySkip	Static	0.794	**0.883**	0.799	0.475
ExerciseOnlySkip	Dynamic	0.805	**0.883**	0.799	0.475

Table 6.1: Mean return of PPO and the baselines in the exercise-only environments. Higher values indicate better overall performance.

As shown in Table 6.1, PPO performs strongly in both exercise-only environments, but it does not outperform the greedy baseline. This pattern is also visible in Figure 6.1, which shows the representative static-pool return comparisons with 95% bootstrap confidence intervals. The greedy baseline outperforming PPO suggests that the exercise-only task is still simple enough that much of the reward structure can be exploited directly by a hand-designed policy. This makes the exercise-only setting an important reference point for the rest of the thesis. It shows that RL already works in this domain, but also that explicit domain knowledge can still be more effective in a simpler setting.

Looking at the individual reward components in Figure 6.2, the behavior of the baselines becomes clearer. The random baseline scores relatively well on uniqueness and the diversity-related components. This is not very surprising, because the exercise catalog is large and a short random session is unlikely to contain exact repeats. Randomly sampled exercises also often differ in their muscle groups. The random policy ignores the session routine, which leads to low routine-match values, and it does not actively avoid overlapping muscle strain, which keeps the fatigue reward low and reduces the overall return.



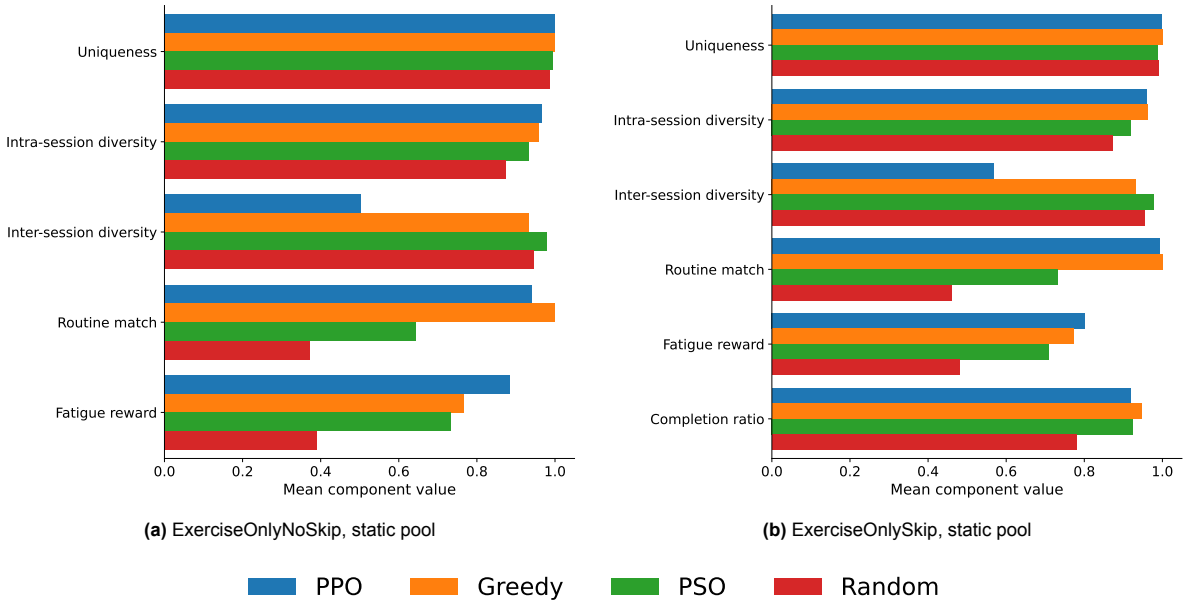
(a) ExerciseOnlyNoSkip, static pool

(b) ExerciseOnlySkip, static pool

Figure 6.1: Representative baseline comparisons in the exercise-only setting using mean return ± 95% bootstrap confidence intervals. Higher return indicates better overall performance. The dynamic-pool comparisons are shown in Appendix B.1.

PSO performs better than random because it searches over complete sessions instead of choosing exercises independently. It evaluates candidate sessions using the terminal reward, which gives it a clear advantage over pure random sampling. PSO is close to PPO, which suggests that offline search over complete sessions can already capture much of the reward structure when the action only consists of choosing exercises. However, PSO still remains below the greedy baseline and does not clearly exceed PPO. In the skip setting, PSO is also limited by the fact that it commits to a full plan at the start and cannot adapt during the episode.

The greedy baseline is not a weak heuristic. It is designed to use much of the same structure that the reward promotes, such as routine alignment, uniqueness, diversity, and strain avoidance. In the implementation, it also includes an explicit fatigue-avoidance term. Its strong performance is therefore not unexpected. This also explains why its routine-match values are almost perfect in both the no-skip and skip settings.



(a) ExerciseOnlyNoSkip, static pool

(b) ExerciseOnlySkip, static pool

Figure 6.2: Representative reward-component means in the exercise-only setting. Higher values indicate better performance for the shown reward components. A shared method legend is shown below the plots. The dynamic-pool versions are shown in Appendix B.2.

Even so, Figure 6.2 shows that PPO performs strongly in this setting. It clearly outperforms random and

remains close to PSO, which shows that reinforcement learning is learning meaningful patterns in the session construction task. At the same time, PPO does not beat greedy on every reward component. In the representative no-skip comparison, PPO achieves a particularly strong fatigue score, but gives up inter-session diversity and a small amount of routine-match performance. This suggests that PPO learns a slightly different trade-off, rather than a uniformly better strategy.

The lower inter-session diversity score should also be interpreted carefully. This component rewards low overlap with the previous session, but in a routine-based gym setting some overlap can be expected when the same routine is sampled repeatedly. In that case, lower inter-session diversity does not necessarily mean that the session is poor; it can also reflect that the policy keeps a consistent routine structure while still controlling fatigue and intra-session diversity. Inter-session diversity is therefore most informative when routines differ between sessions, where repeated muscle overlap is less desirable.

The same overall picture remains in the skip setting, as shown in Figure 6.2. Greedy and PPO both keep completion high, while the random baseline performs much worse. Since the skip environments scale the final reward by completion ratio, this directly explains part of the return gap. PPO is therefore able to learn skip-aware behavior, but the greedy baseline still performs slightly better overall in the exercise-only case.

Overall, this subsection shows that PPO learns meaningful exercise-only recommendation behavior, but does not clearly dominate the stronger non-learning baselines. It clearly improves over random recommendation and remains competitive with PSO, but a strong reward-aligned heuristic still performs best. This suggests that, when only exercise identities are selected, reinforcement learning is not yet necessary to outperform the best hand-crafted baseline. The main point is therefore not that RL fails, but that the exercise-only action space is still too limited for its added complexity to fully pay off. This motivates the move to the full-prescription setting in the next subsection.

6.2. RL performance in full-prescription recommendation

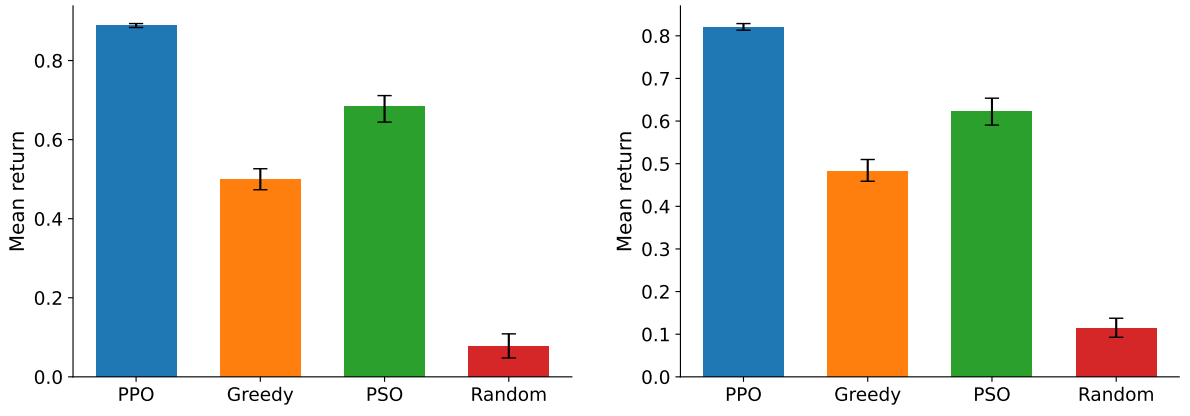
As shown in Table 6.2, PPO outperforms all baselines in the full-prescription environments. This pattern is also visible in Figure 6.3, which shows representative static-pool return comparisons with 95% bootstrap confidence intervals. The gap between PPO and the non-learning baselines is clear, especially compared with greedy and random, and PSO also remains below PPO. Most importantly, the greedy baseline, which was strongest in the exercise-only setting, is no longer competitive here.

This is a clear change from Section 6.1. In the exercise-only setting, PPO learned useful structure but still remained below greedy. In the full-prescription environments, that ranking changes completely. PPO becomes the best method, while greedy loses its dominant position and, in the representative no-skip comparison, even performs worse than PSO. This shows that the real difficulty of the task is both choosing a set of exercises and choosing a suitable prescription for each exercise.

Environment	Pool	PPO	Greedy	PSO	Random
FullPrescriptionNoSkip	Static	**0.889**	0.500	0.683	0.077
FullPrescriptionNoSkip	Dynamic	**0.892**	0.500	0.683	0.077
FullPrescriptionSkip	Static	**0.821**	0.483	0.623	0.115
FullPrescriptionSkip	Dynamic	**0.823**	0.483	0.623	0.115

Table 6.2: Mean return of PPO and the baselines in the full-prescription environments. Higher values indicate better overall performance.

This difference can be understood through the reward design. In the full-prescription environments, the agent is no longer evaluated only on sequence-level qualities such as routine match, uniqueness, and diversity. It is also rewarded for prescription quality. This means that each exercise must be paired with appropriate sets, reps, and load for the user. In addition, the prescription reward includes a weakest-link effect, so a single poor prescription can noticeably lower the session score. As a result, simple exercise-level heuristics are no longer enough. The policy must make good exercise and prescription decisions together.



(a) FullPrescriptionNoSkip, static pool

(b) FullPrescriptionSkip, static pool

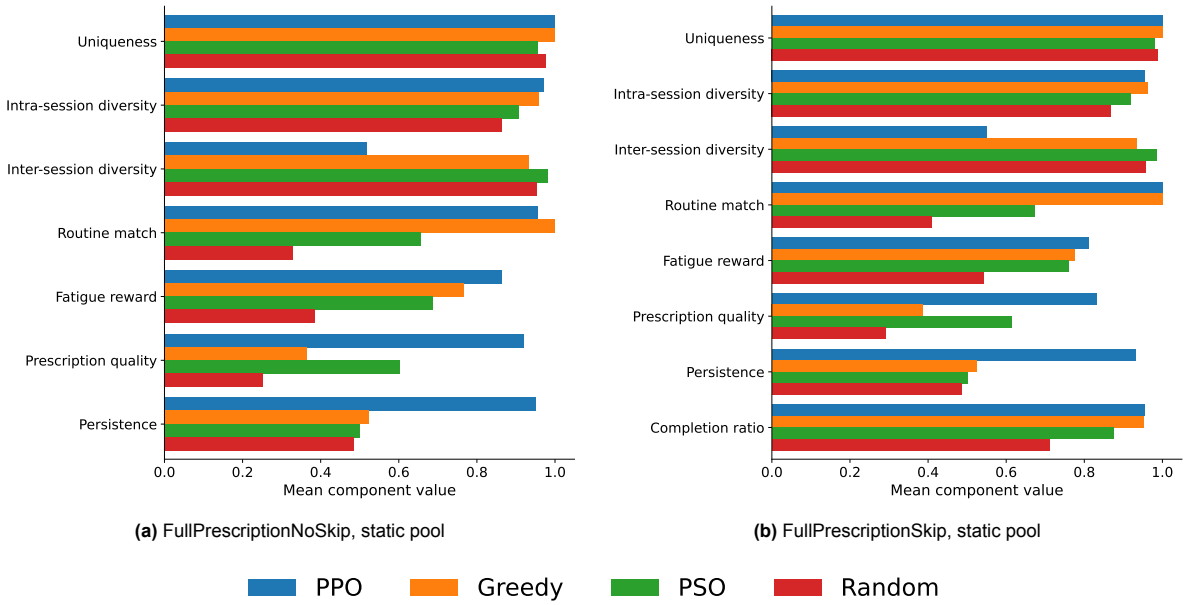
Figure 6.3: Representative baseline comparisons in the full-prescription setting using mean return $\pm$ 95% bootstrap confidence intervals. Higher return indicates better overall performance. The dynamic-pool comparisons are shown in Appendix B.1.

The reward-component plots in Figure 6.4 make the differences between baselines clearer. As in the exercise-only setting, the random baseline still scores reasonably on some sequence-level components, especially uniqueness and diversity. This is again mostly due to the large catalog. However, once sets, repetitions, and load are included, random selection becomes much less effective. The prescription-quality component is very low, which shows that even if a random session looks diverse, the actual prescriptions are usually implausible or poorly matched to the user. In the skip setting, this becomes even more visible because poor prescriptions lower completion and increase skipping.

The greedy baseline fails for a different reason. It still performs well on the sequence-based components, especially routine match and diversity, because it was explicitly built to align with those parts of the reward. However, its prescription rules are fairly fixed and depend mostly on goal and experience, with only limited adjustment for recovery or workload. In other words, greedy remains a strong session-building heuristic, but it is not a truly personalized prescription model. This can also be seen in Figure 6.4: routine-match scores remain very high, but prescription-quality scores are much lower than those of PPO. Greedy therefore produces sessions that look structurally good, but are often not the best prescriptions for the specific user.

PSO lies somewhere in between. Unlike greedy, it searches over full session plans and therefore has a better chance of finding sessions that score well under the terminal reward. This likely explains why PSO performs much better than greedy on the prescription-quality term in the representative no-skip setting. Still, PSO remains below PPO overall. A likely reason is that PSO performs a limited offline search. It evaluates a finite number of candidate sessions and then commits to the best one it finds. This can improve over fixed rules, but it does not provide the same policy learning across users as PPO. In the skip setting, PSO is also at a disadvantage because it plans at the start of the episode and cannot adapt to skip outcomes during the session.

Figure 6.4 suggests that PPO performs best because it learns the joint structure of the problem instead of separating exercise choice from prescription choice. The component plots suggest that PPO is not simply maximizing every reward term independently. Rather, it learns a better overall trade-off: routine match stays high, fatigue remains well controlled, persistence is strongest, and prescription quality is much higher than for the baselines. PPO often gives up some inter-session diversity compared with PSO or greedy, but this appears to be a trade-off rather than a weakness. This is especially important to interpret correctly in the full-prescription setting. Low inter-session diversity means that the current session overlaps more with the previous session in terms of trained muscles. That can be undesirable when the routine changes, but it is not automatically bad when the same routine is sampled again. In that case, some overlap is expected, and the policy may prioritize routine consistency, persistence, prescription quality, and fatigue management over maximizing inter-session novelty. The policy is willing to keep useful structure across sessions when that helps it achieve better prescriptions and higher overall return.



(a) FullPrescriptionNoSkip, static pool

(b) FullPrescriptionSkip, static pool

Figure 6.4: Representative reward-component means in the full-prescription setting. Higher values indicate better performance for the shown reward components. A shared method legend is shown below the plots. The dynamic-pool versions are shown in Appendix B.2.

This advantage is especially visible in the full-prescription skip environment in Figures 6.3 and 6.4. In this setting, the policy must recommend prescriptions that score well under the reward, but also prescriptions that users actually complete. Because the final reward is scaled by completion ratio, poor adherence directly lowers performance. The component plots show that PPO combines the highest prescription-quality score with a high completion ratio. This suggests that PPO is learning prescriptions that are good in theory, but also practical under the simulator dynamics.

Taken together, these findings show that reinforcement learning may not be needed in simplified exercise-selection settings, but becomes clearly more useful when the problem is extended to realistic full-prescription recommendation. The main advantage of RL in this setting is therefore not just sequencing, but sequencing under personalized prescription constraints.

6.3. Static versus dynamic user pools

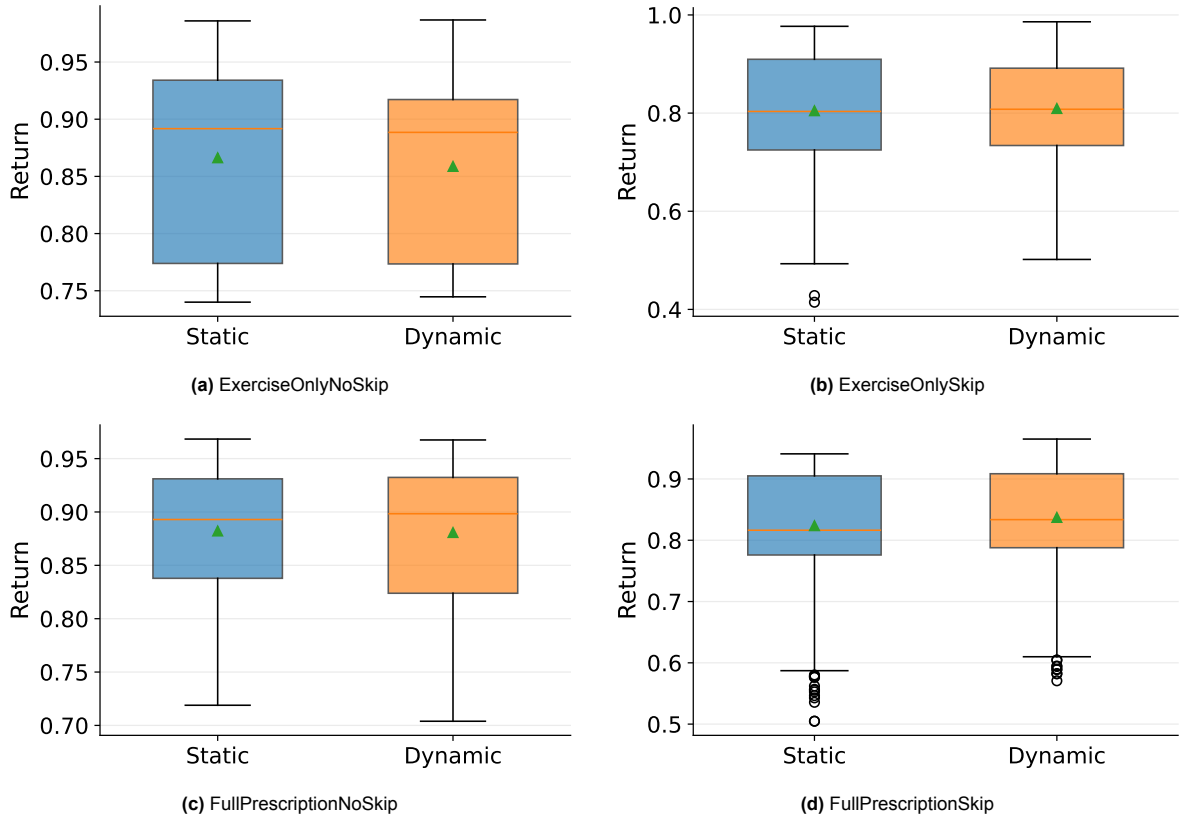
This subsection compares PPO models trained on static user pools with PPO models trained on user pools that change gradually over time. The main question is whether moderate user drift changes the learning process or lowers the final recommendation quality.

Figure 6.5 compares PPO performance under static and dynamic user pools across all four environments. In the boxplots, the horizontal line denotes the median episode return, the green triangle denotes the mean, and the box shows the interquartile range. When comparing static and dynamic user pools, PPO reaches very similar final performance in both settings. Moving from a stationary pool to an evolving one does not lead to a clear drop in return. Some environments show a small increase under the dynamic pool, while others show a small decrease, but the differences are limited in all cases.

The training curves in Figure 6.7 support the same conclusion. The static and dynamic runs are not exactly the same, but they follow the same general pattern and converge to similar final plateaus. In all four environments, return drops or fluctuates a bit at the start of training and then increases steadily. The dynamic pools do not lead to visible instability, divergence, or much slower convergence. This suggests that the level of non-stationarity introduced by gradual user evolution is still mild enough for PPO to handle well.

The component-level plots in Figure 6.6 show the same picture. Across all four environments, the static and dynamic component profiles are very similar. The differences are small and mostly appear as minor

shifts between inter-session diversity, fatigue, prescription quality, persistence, and completion ratio. In some cases the dynamic pool is slightly higher, while in other cases the static pool is slightly higher. The important point is therefore not the direction of each individual component change, but the absence of a large degradation when users evolve gradually over time.



(a) ExerciseOnlyNoSkip **(b)** ExerciseOnlySkip

(c) FullPrescriptionNoSkip **(d)** FullPrescriptionSkip

Figure 6.5: Comparison of PPO performance under static and dynamic user pools across all four environments using episode-level return distributions. Higher return indicates better overall performance. The boxplots show that the static and dynamic distributions overlap strongly in all cases, indicating that moderate user drift does not materially reduce final return.

A more careful conclusion is therefore that PPO learns policies that remain effective under gradual changes in user state over time. Instead of overfitting to a completely stationary training distribution, the learned policy seems robust to the level of drift introduced by the dynamic pools.

At the same time, this result should be interpreted together with the way the dynamic pools are defined. The drift is intentionally moderate and realistic. Users change between sessions through a gap model and small long-term updates, not through abrupt or extreme changes. Under these conditions, many user features stay close to their earlier values, so the best recommendation does not need to change dramatically from one session to the next. The more difficult question is whether PPO stays robust when the drift becomes deliberately unrealistic, which is examined in Section 6.5.

Overall, the results indicate that the framework remains stable under realistic, gradual user drift. Replacing a static user pool with an evolving one does not materially lower PPO performance, and it also does not substantially change the learned reward trade-offs.

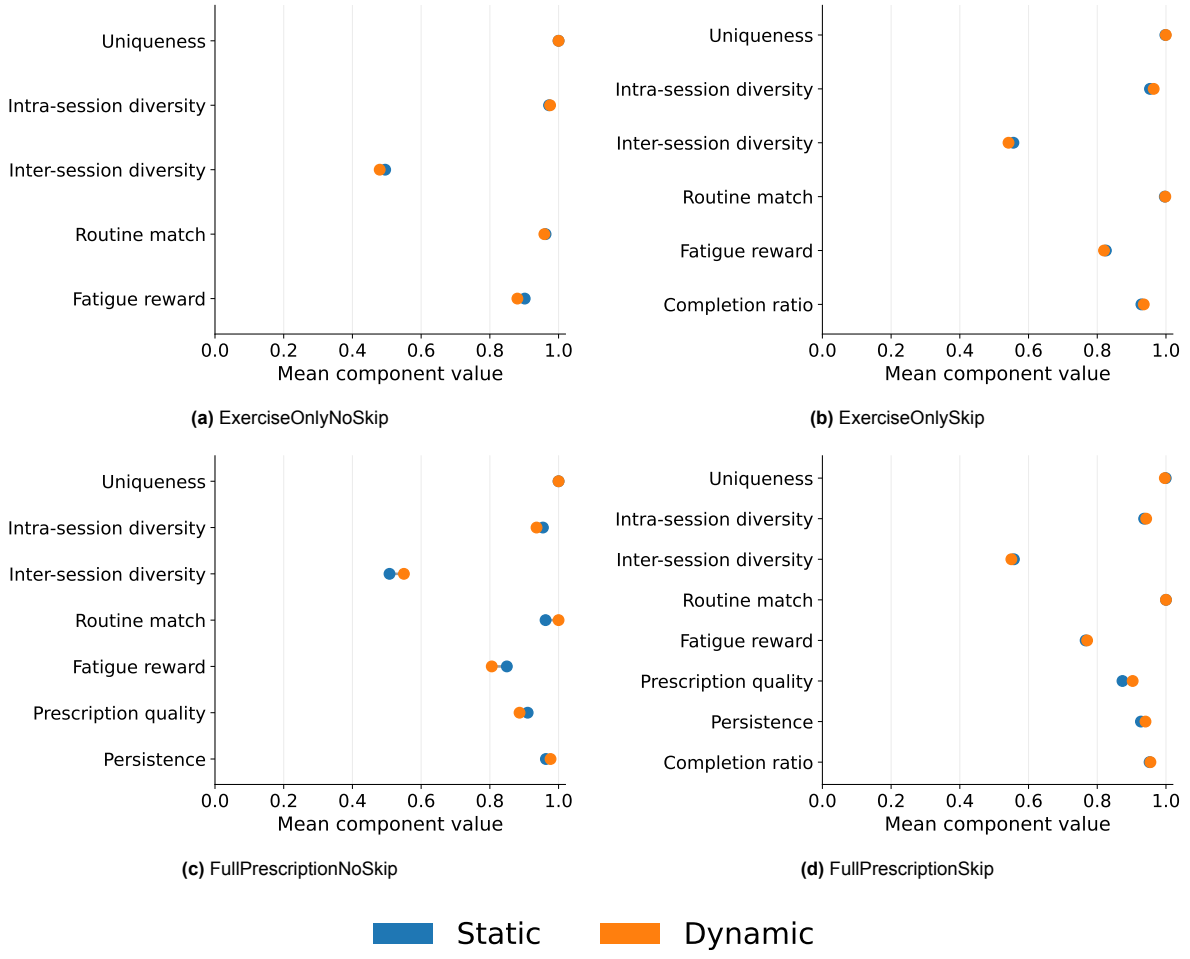


Figure 6.6: Component means of PPO performance under static and dynamic user pools across all four environments. Higher values indicate better performance for the shown components. The nearby static and dynamic points show that gradual user drift leads only to small component-level changes. A shared color legend is shown below the figure.

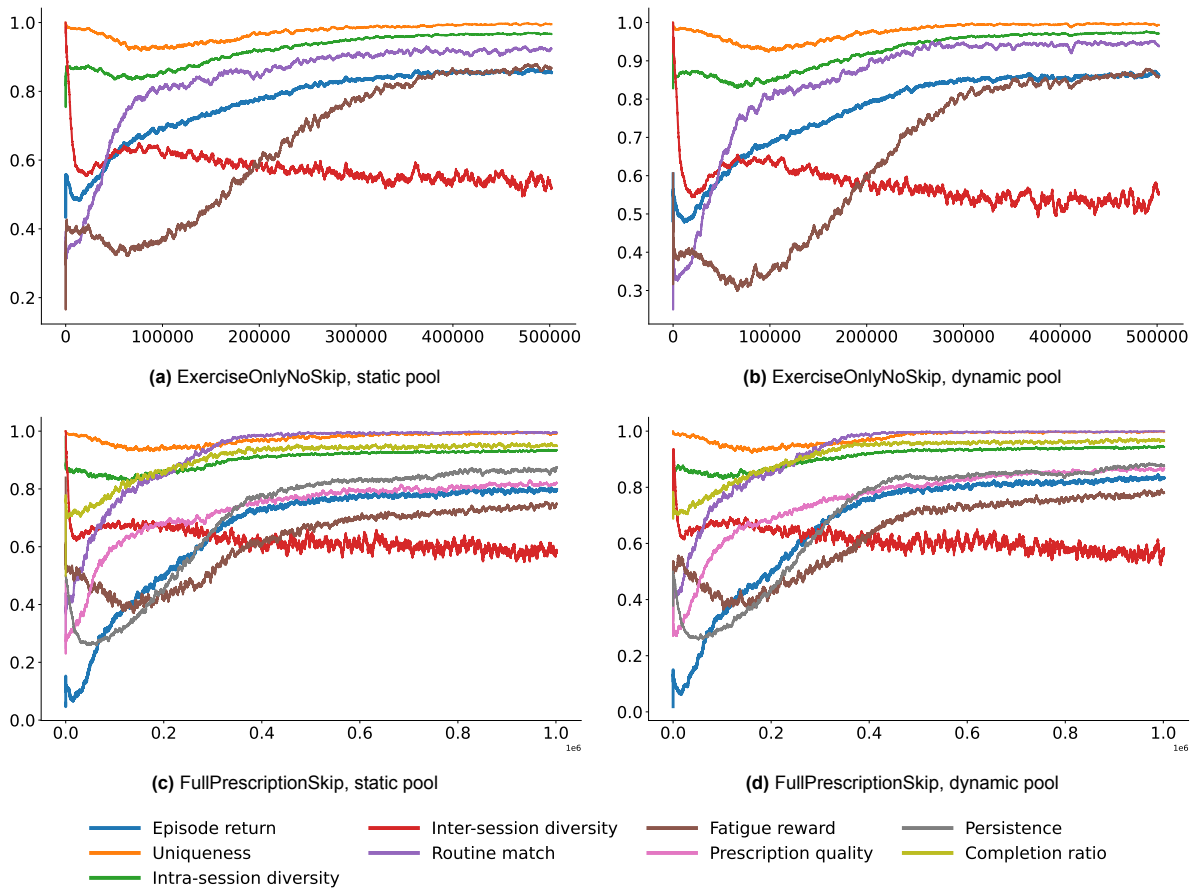


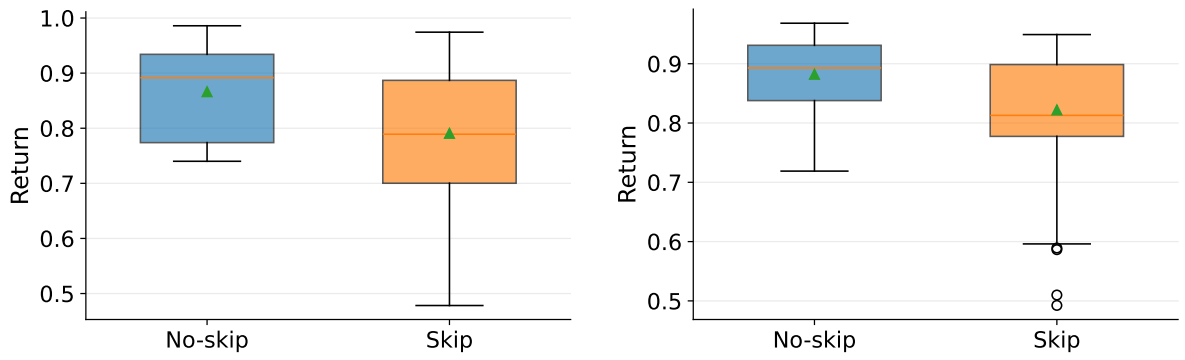
Figure 6.7: Representative PPO training curves under static and dynamic user pools. In both the simpler exercise-only setting and the harder full-prescription skip setting, the learning trajectories remain stable and converge to similar final values. Shared color legends are shown below each row. The remaining training curves are shown in Appendix B.3.

6.4. Effect of skipping and online personalization

Figures 6.8–6.10 compare PPO performance between no-skip and skip-enabled environments. The skip-enabled environments differ from the no-skip environments in two main ways. First, the terminal reward is computed on the completed session instead of the suggested session, and it is scaled by completion ratio. This means the policy is directly rewarded for recommendations that users actually perform. Second, skip outcomes are used as skip-only feedback to update the online user model during the episode. This allows the policy to adapt later actions based on estimated tolerance and avoidance patterns. The results in this subsection should therefore be interpreted as the combined effect of skipping and online personalization, rather than skipping alone.

As shown in Figure 6.9, the clearest difference appears in the routine-match component. In both the exercise-only and full-prescription settings, routine-match scores are higher in the skip-enabled environments than in the no-skip variants. This fits the simulator design. Recommendations that do not match the intended routine are more likely to be skipped, and because the reward is then computed on the completed session, the learner receives a stronger signal in favor of routine-consistent recommendations.

A second consistent difference in Figure 6.9 appears in the fatigue component, which tends to be lower in the skip environments. This can be explained by the way the reward is computed. In the skip variants, fatigue is evaluated on the performed session rather than the suggested one. When a user skips an exercise in the middle of the sequence, the performed session becomes shorter and some muscle exposures move closer together. As a result, a session that was reasonably spaced when suggested can still end up with a lower fatigue score after it is evaluated on the completed sequence. This effect becomes stronger because skip-enabled models are also pushed toward stronger routine consistency, which can increase overlap within the same muscle family.



(a) Exercise-only, static pool

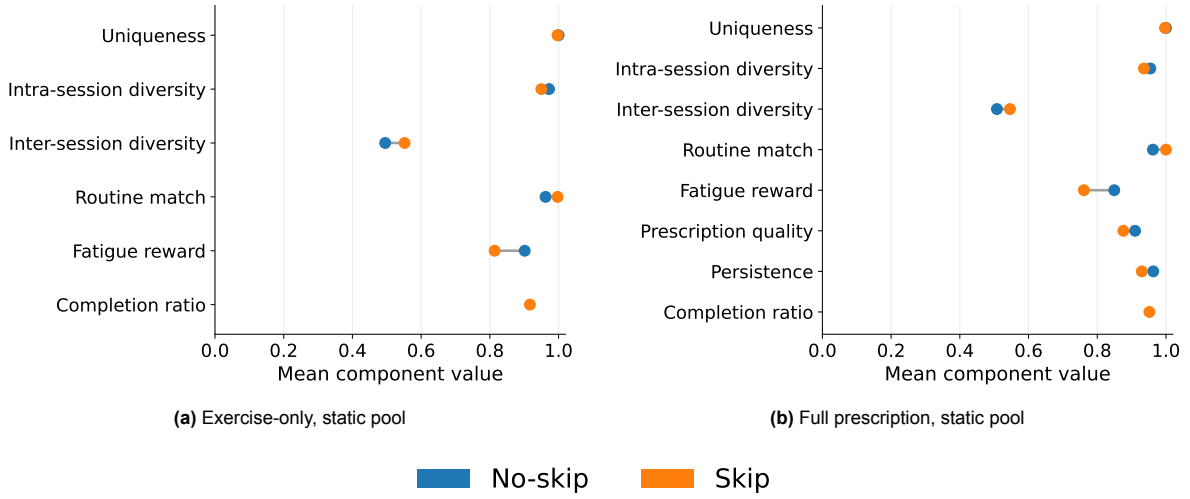
(b) Full prescription, static pool

Figure 6.8: Return comparison between no-skip and skip-enabled PPO environments. Higher return indicates better overall performance. In both action spaces, final return is slightly lower in the skip setting because the reward is based on the completed session and scaled by adherence.

Figure 6.8 shows that the overall return in the skip environments is slightly lower than in the no-skip variants. This is expected, because the final reward is scaled by completion. The training curves in Figure 6.10 show that the completion ratio increases clearly during learning and then converges to a high score close to one. In the full-prescription environments this level is close to the base skip noise built into the simulator, while in the exercise-only environments it stays slightly higher. This suggests that PPO learns to reduce most avoidable skipping, but cannot reduce it fully to zero because some skipping is part of the simulator noise and some still comes from mismatch between recommendations and user state.

This trade-off is most visible in the full-prescription setting in Figure 6.9. Compared with the no-skip model, the skip-enabled model reaches stronger routine-match behavior and maintains a high completion ratio, but it converges to somewhat lower fatigue and prescription-quality scores. This suggests that once adherence becomes part of the objective, PPO is no longer only optimizing for the best prescription on paper. Instead, it learns a more conservative and adherence-aware policy that gives

up some session-quality components in order to keep recommendations executable under uncertainty about the user.



(a) Exercise-only, static pool

(b) Full prescription, static pool

Figure 6.9: Component means for no-skip and skip-enabled PPO environments. Higher values indicate better performance for the shown reward components. The paired points show how enabling skipping changes the component profile, especially routine alignment, fatigue, and prescription quality. A shared color legend is shown below the figure.

The training curves in Figure 6.10 show this difference more clearly than the final summary plots alone. In the exercise-only setting, the skip-enabled models increase routine match more quickly and reach a higher final level than the no-skip models, while fatigue improves more slowly and ends at a lower plateau. In the full-prescription setting, the same pattern appears but more strongly. Routine match and adherence-related quantities improve quickly, and completion ratio increases sharply during training. At the same time, fatigue and prescription quality improve more gradually and remain somewhat below the no-skip variant. This suggests that skip-based interaction changes both the final scores and also which reward components become useful earlier in learning.

Overall, the skip-enabled environments show that implicit user feedback changes the optimization problem in a meaningful way. PPO learns to construct high-quality sessions while also keeping those sessions performable after user interaction. The result is not necessarily a higher final reward than in the no-skip case, but a more adherence-aware policy that better reflects the practical constraints of real-world recommendation.

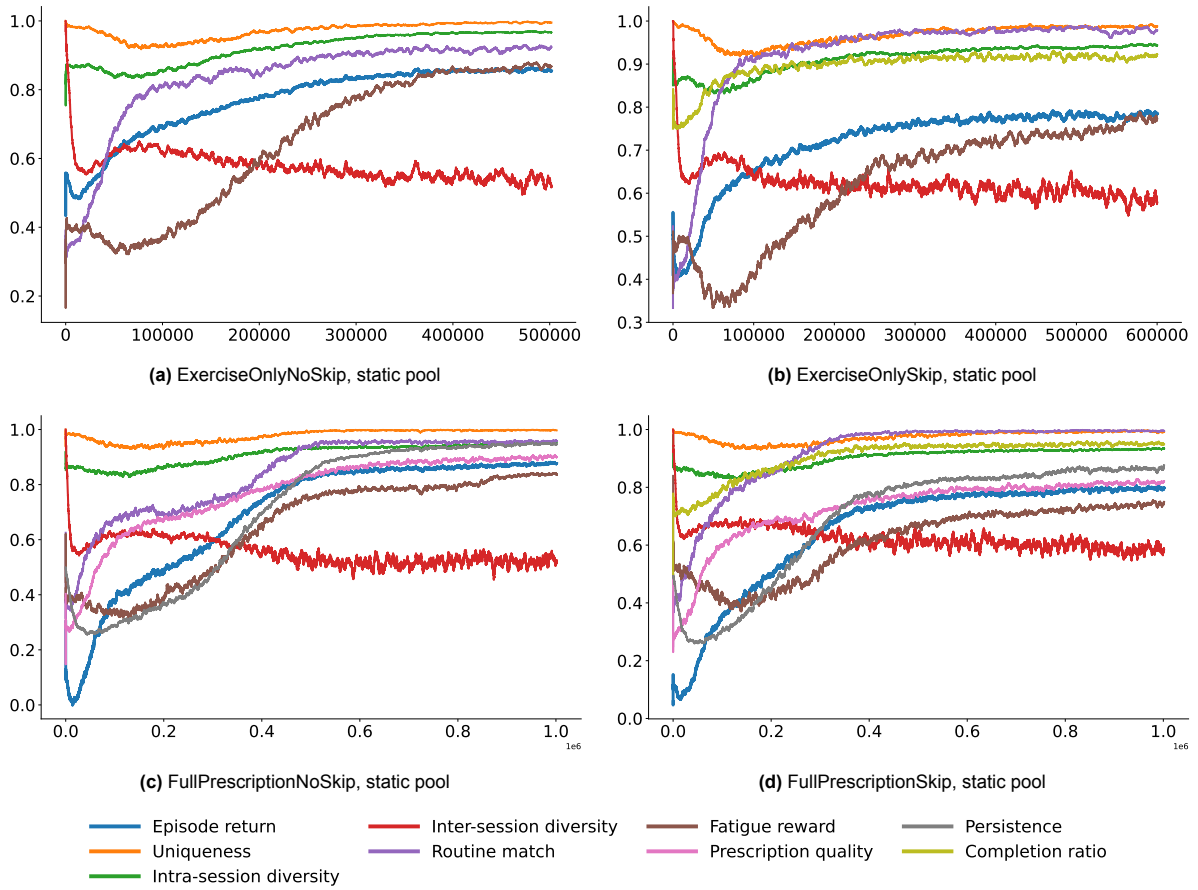


Figure 6.10: Representative PPO training curves comparing no-skip and skip-enabled environments for the static pool. Skip-enabled learning leads to faster convergence of routine-related behavior and stronger adherence signals, while fatigue and, in full prescription, prescription-quality components plateau somewhat lower. Shared color legends are shown below each row. The dynamic-pool versions are shown in Appendix B.4.

6.5. Robustness under extreme user drift

This subsection studies robustness under two deliberately unrealistic stress-test settings: a robust dynamic pool, where user variables drift strongly but in a consistent direction, and a chaotic dynamic pool, where user variables move randomly up and down between sessions. These settings are meant to test the limits of the framework under much stronger distribution shifts than those in the standard dynamic pools.

Figure 6.11 shows that the robust dynamic regime has only a small effect on final PPO performance. Across both action spaces, return stays close to the standard dynamic setting. Some comparisons show a small improvement under the robust pool, while others show a small decrease, but there is no indication of policy collapse. This suggests that large drift by itself is not necessarily the hardest part. As long as the direction of change remains structured, PPO and the online state representation can still track the environment reasonably well.

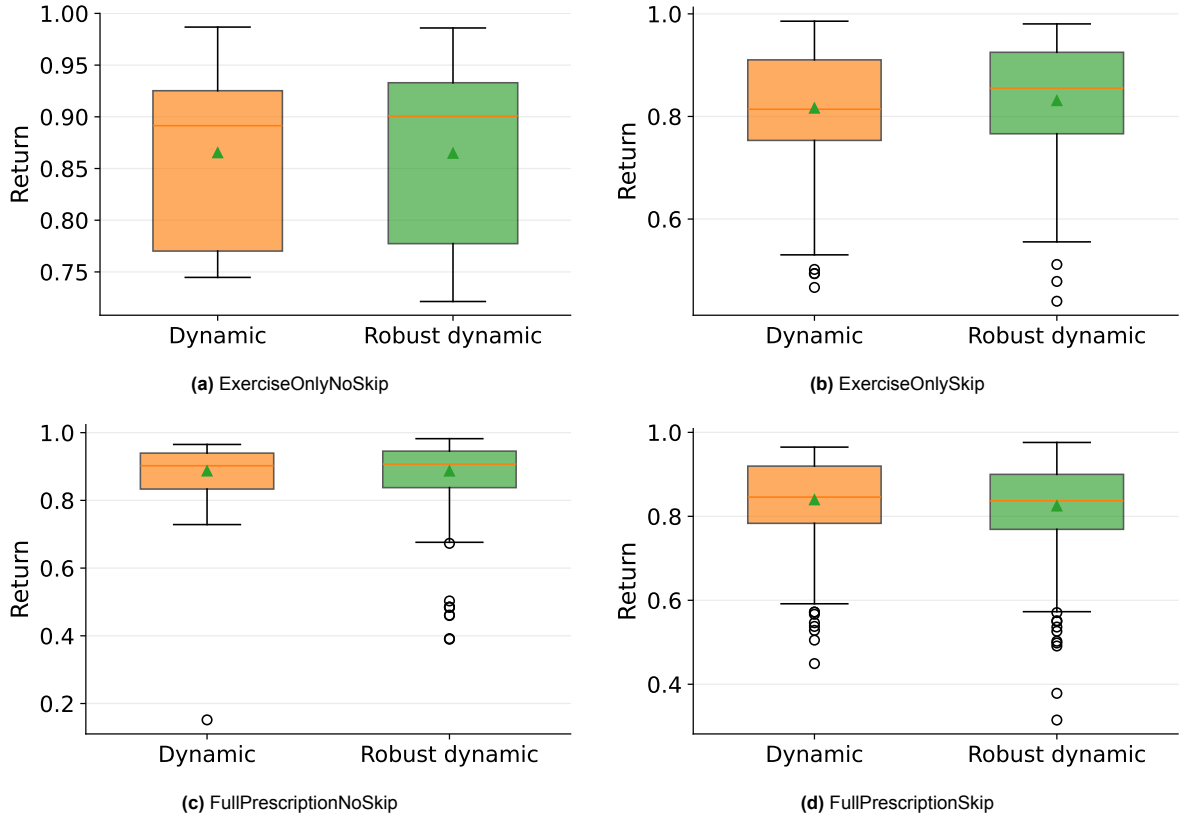
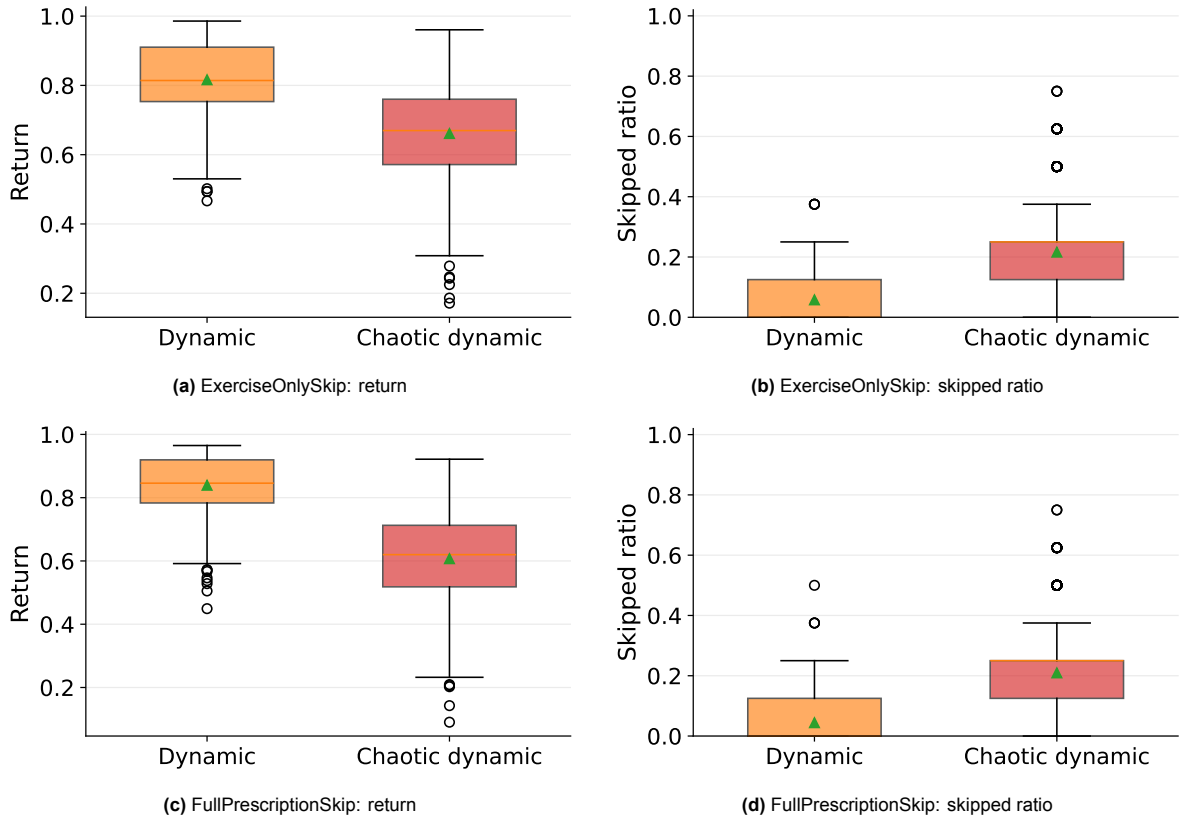


Figure 6.11: Comparison between the standard dynamic pool and the robust dynamic stress test using episode-level return distributions. Higher return indicates better overall performance. Strong but directional drift leads to only limited changes in PPO performance.

The chaotic dynamic regime gives a different result. The training curves in Figure 6.13 and Appendix B.5 suggest that degradation is more limited in the no-skip environments than in the skip-enabled ones. In the exercise-only no-skip case, performance remains close to the earlier plateau, while in the full-prescription no-skip case the decrease is visible but still limited. One explanation is that, without skipping, the policy is not directly penalized through adherence loss or stale skip-based beliefs. The chaotic user evolution may therefore average out enough that a policy optimized for the population mean can still perform reasonably well.

Figure 6.12 shows a different result for the skip-enabled environments: the sharp increase in skipped ratio is accompanied by a substantial drop in overall return. The policy now has to do more than recommend a good session. It also has to maintain useful online beliefs about user tolerance and avoidance. When the underlying user variables change randomly between sessions, these beliefs become outdated very quickly. The result is a strong increase in skipped ratio and a corresponding

decrease in completion. Since the final reward in skip environments is scaled by adherence, this directly lowers return. This effect is visible in both exercise-only skip and full-prescription skip, with the largest return drop appearing in the full-prescription skip case.



(a) ExerciseOnlySkip: return

(b) ExerciseOnlySkip: skipped ratio

(c) FullPrescriptionSkip: return

(d) FullPrescriptionSkip: skipped ratio

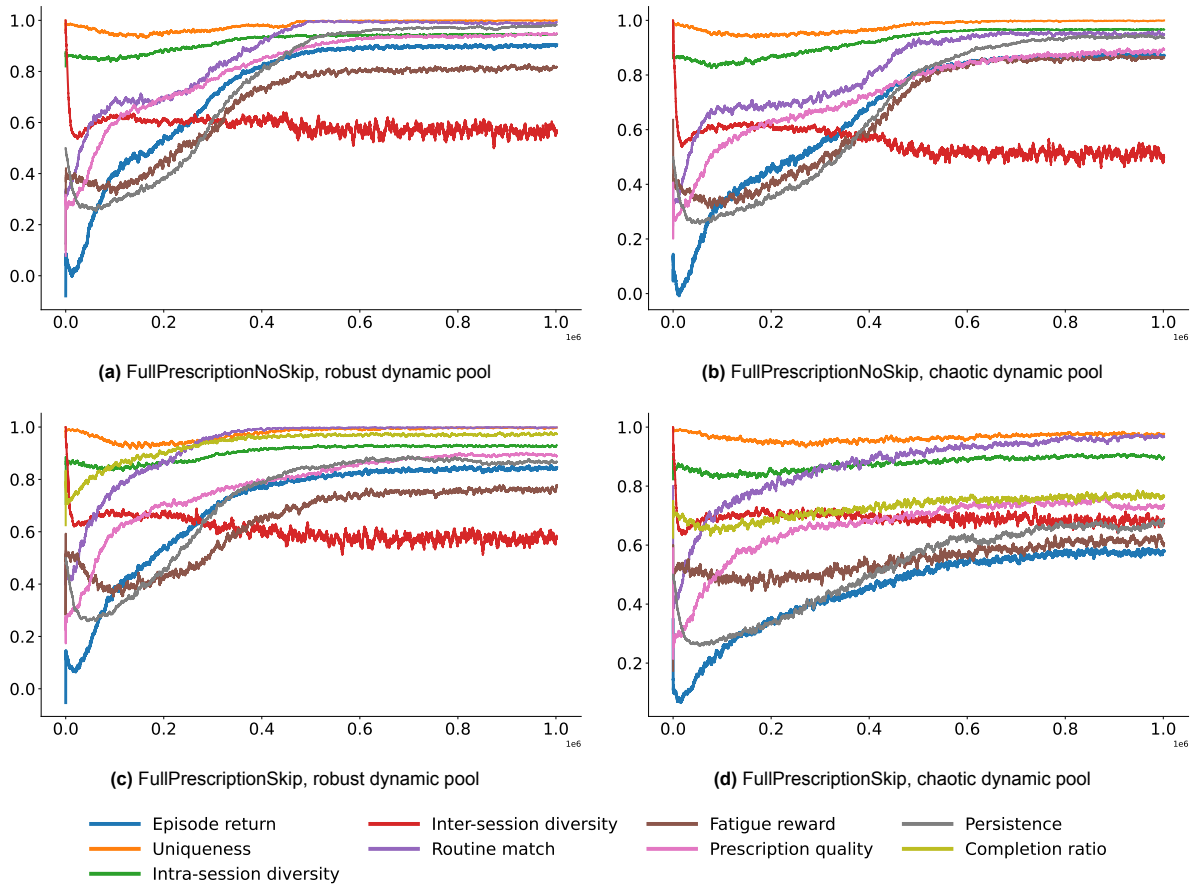
Figure 6.12: Effect of chaotic drift in skip-enabled environments. Higher return indicates better overall performance, while lower skipped ratio indicates better adherence. The sharp increase in skipped ratio is accompanied by a substantial drop in overall return.

The training curves in Figure 6.13 make the contrast between robust and chaotic drift clearer. Under robust dynamic drift, PPO still converges in a stable way. Return increases steadily, adherence-related terms remain strong, and in the skip-enabled environments the completion ratio still increases to a high value. Under chaotic drift, the learning dynamics change much more. In the no-skip environments, PPO still converges, but to a lower plateau. In the skip-enabled environments, the degradation is much stronger: completion ratio stays lower and return converges to a much worse value. At the same time, structural components such as uniqueness and routine match still become high. This shows that chaotic drift does not stop learning completely, but it does break the usefulness of online personalization and adherence-sensitive decision making.

The full-prescription skip environment is the clearest stress test in this chapter. Figures 6.12 and 6.13 show that, under chaotic drift, completion ratio stays relatively low and overall performance deteriorates most clearly in this setting. This suggests that the model can no longer carry useful information from one session to the next. The personalized beliefs formed online become outdated too quickly for consistent prescription planning to remain effective.

Taken together, these stress tests show that the framework is robust to large but structured drift, but much less robust to highly non-stationary and rapidly reversing user dynamics. The contrast between robust dynamic and chaotic dynamic suggests that unpredictability matters more than the size of the drift. Strong but consistent drift can still be tracked, whereas rapidly changing drift breaks the usefulness of online personalization.

These extreme settings are intentionally unrealistic and mainly serve as stress tests. The results should therefore be read as showing the limits of the approach, rather than its expected performance in real-



(a) FullPrescriptionNoSkip, robust dynamic pool

(b) FullPrescriptionNoSkip, chaotic dynamic pool

(c) FullPrescriptionSkip, robust dynamic pool

(d) FullPrescriptionSkip, chaotic dynamic pool

Figure 6.13: Representative training curves for the robustness stress tests in the full-prescription environments. Robust dynamic drift still allows stable convergence, whereas chaotic drift mainly disrupts adherence-sensitive and personalization-dependent components, especially in the skip-enabled setting. Shared color legends are shown below each row. The exercise-only versions are shown in Appendix B.5.

world use.

7

Discussion

This chapter discusses the main findings of the thesis and places them in the broader context of the research process, the methodological choices, and the intended contribution of this work. The discussion focuses on the final numerical results, but also on why the thesis was designed in this way, what the results mean for gym workout recommendation, and where the current approach still has clear limitations.

7.1. From early problem exploration to the final thesis design

An important part of this thesis is that the final system design did not arise immediately, but was shaped by the constraints and gaps found during the first phase of the project. Early in the thesis, several directions were considered for creating a more tangible exercise recommender, including ideas based on video, sensory data, and exercise-quality assessment. These directions were appealing because they could potentially provide direct and interpretable feedback to users. However, the literature review and data search showed that these data sources mostly support recognition, repetition counting, or form analysis, rather than sequential gym session recommendation. In particular, they rarely contain the type of user-session interaction data needed to train and evaluate a system that recommends full workout sessions over time.

This was an important turning point in the thesis. The problem was no longer simply how to build a gym recommender, but how to study gym recommendation in a setting where the needed interaction data is largely unavailable. The eventual use of synthetic users and a simulator followed directly from the practical lack of suitable real-world gym recommendation data.

The HFRL framework then became the most relevant starting point. It offered a concrete example of workout recommendation as a sequential RL problem, together with a user simulator and a session-level reward design. At the same time, the paper also left several implementation details underspecified, including parts of the input encoding, reward weighting, and skip-related behavior. This meant that the thesis could not treat the HFRL framework as a line-by-line reproduction target. Instead, it had to be treated as a conceptual foundation that required explicit interpretation and extension.

That observation influenced the thesis in two ways. First, it justified beginning with an HFRL-style exercise-only setting, because this provided the clearest way to connect back to the original framework. Second, it also motivated going beyond that setting, because exact reproduction alone would not have answered the main research question of this thesis. The central interest here was both whether RL can recommend exercises and whether RL becomes more useful when the gym recommendation problem becomes more realistic, more sequential, and more personalized.

In that sense, the final thesis design reflects the full path of the project. The work started with a broad exploration of physical activity recommender systems, wearable and video-based feedback, and data availability, and gradually narrowed toward a simulator-based RL setup for gym sessions. From there, the thesis moved from a simpler exercise-only formulation to a more complex full-prescription formula-

tion, and from static recommendation to recommendation under skip-based interaction and user change over time. This progression gives the thesis its main structure, and the discussion of the results should be interpreted in light of that progression.

7.2. Interpretation of the main findings

The main result of the thesis is not that reinforcement learning is always better than non-learning methods. A more accurate interpretation is that the value of RL depends strongly on the complexity of the recommendation problem. This is the most important pattern that appears across the full set of experiments.

In the exercise-only environments, PPO performs well, but it does not outperform the greedy baseline. This is an important finding, because it shows that in the simpler HFRL-style setting, much of the useful session structure can still be captured by a strong hand-crafted policy. In other words, once the recommendation task is limited to selecting only exercise identities, a heuristic that directly follows the reward structure can already perform very strongly. This does not mean that RL fails. PPO still clearly outperforms the weaker baselines, which shows that it learns meaningful structure. However, it does mean that the exercise-only setting is not yet complex enough to clearly justify RL over a good heuristic.

This finding is actually valuable for the thesis, because it prevents the argument from becoming too strong or too convenient. If PPO had outperformed everything in every setting, the work would have had a simpler story, but also a less informative one. The fact that greedy remains best in the exercise-only environments makes the later full-prescription result much more convincing. It shows that the thesis does not begin with the assumption that RL must win, but instead identifies more precisely where the advantage of RL starts to appear.

That advantage becomes clear in the full-prescription environments. Here, PPO outperforms all baselines, including greedy. This is the strongest result of the thesis, because it directly supports the idea that gym recommendation becomes a substantially harder planning problem once the system must recommend which exercise to do, but also how to do it. Sets, repetitions, and load change the problem from sequence construction to joint sequence-and-prescription construction. A recommendation can no longer be judged only by whether the chosen exercises look diverse or match the intended routine. It must also be judged by whether the prescribed effort is appropriate for the user.

This is also where the role of hidden user state becomes more important. In the full-prescription environments, the baselines do not only have to choose plausible exercises. They also have to estimate what a user can handle. The prescription-quality reward and the skip-related dynamics together make the task more sensitive to personalized capacity and tolerance. That seems to be the point where PPO gains a clear advantage. The learned policy can adapt to the structure of the environment and the interaction patterns in a way that the simpler baselines cannot. This is why the main contribution of RL in this thesis is not just sequencing, but personalized sequencing under prescription constraints.

The skip-enabled environments provide a second important insight. They show that adding user interaction changes the optimization problem itself. In the no-skip environments, the model is rewarded for constructing good sessions on paper. In the skip environments, the model is rewarded for constructing sessions that are actually completed. This is an important conceptual shift. The recommendation system is no longer optimizing only for abstract session quality, but also for adherence.

The results suggest that this shift has meaningful effects. Skip-enabled models tend to learn stronger routine-consistent behavior, and in the full-prescription setting they also learn to keep skipped ratios low while maintaining high completion. At the same time, skip-enabled learning introduces visible trade-offs. Some components, especially fatigue and in some cases prescription quality, end at slightly lower values than in the no-skip setting. This should not be interpreted as a weakness of the skip environments. Instead, it suggests that the model is learning a different objective: not the theoretically best session, but the best session that the user is likely to perform. In practical recommendation settings, that is arguably the more relevant objective.

The static versus dynamic comparison adds another layer to the discussion. Here, the main result is that PPO remains stable under gradual and realistic user drift. This is important because it suggests that the model is not simply memorizing a fixed population. It can still learn effectively when users

evolve slowly over time. At the same time, this result should be interpreted carefully. The dynamic pools are intentionally moderate. Users change, but they do not change drastically from one session to the next. That makes this result encouraging, but it should not be overstated.

This is exactly why the robustness experiments are useful. They show that the framework can still handle large but structured drift reasonably well, but becomes much more fragile when drift is chaotic and rapidly reversing. This result is especially meaningful in the skip-enabled environments. When the environment changes unpredictably, the online beliefs built from skip feedback become stale very quickly. The model still learns some structural aspects of the task, but its adherence-sensitive behavior breaks down. This suggests that, for online personalization, unpredictability matters more than the size of the change itself. Large change can still be tracked when it is coherent. Random reversal is much harder.

Taken together, these findings clarify more precisely what RL contributes in this thesis. RL is not necessary in the simplest exercise-sequencing setting, where a strong heuristic can already perform very well. Its value becomes much clearer once the recommendation problem includes prescription complexity, user-specific tolerance, adherence-sensitive reward, and interaction over time. This makes the contribution of RL in this work more specific than a general claim that RL is always better for workout recommendation.

7.3. Discussion in relation to the research questions

The results discussed in this chapter provide clear answers to the research questions introduced in Chapter 1.

Sub-RQ1 concerned the HFRL-style exercise-selection setting. The results show that PPO clearly outperforms PSO and random, but not the greedy baseline. This means that the qualitative result of RL beating weaker baselines is reproduced, while also showing that a strong heuristic remains very competitive in the gym-domain exercise-only case.

Sub-RQ2 asked whether full prescription creates conditions in which RL reliably outperforms non-RL baselines. The answer is yes. Once the recommendation includes sets, repetitions, and load, PPO becomes the strongest method across the baseline comparisons. This is the clearest direct support for the main claim of the thesis.

Sub-RQ3 addressed skipping and online personalization from skip-only feedback. The main conclusion is that skipping changes the recommendation objective in a meaningful way. The skip-enabled environments encourage more adherence-aware behavior, but also introduce trade-offs in other reward components.

Sub-RQ4 concerned dynamic user pools and robustness. The results show that PPO handles realistic gradual user drift well, but that highly chaotic user change breaks the usefulness of online personalization, especially in the skip-enabled environments.

Taken together, the research questions are answered in a consistent way. Reinforcement learning is not shown to be uniformly superior in all workout recommendation settings, but it does become clearly more useful once the problem includes full prescription, user interaction, and personalization under changing conditions.

7.4. Implications for gym workout recommendation

Beyond the direct thesis results, the findings also have some broader implications for how gym workout recommenders might be designed in practice.

A first implication is that not every gym recommendation problem needs RL. If the task is mainly to select exercises that fit a routine and maintain reasonable diversity, then a strong heuristic may already be enough. This is an important practical point, because RL introduces more engineering complexity, training cost, and interpretability challenges. In such simpler settings, a carefully designed heuristic may be easier to deploy and justify.

A second implication is that the real value of RL appears once gym recommendation includes dose and

adherence. In practical gym systems, users are rarely only asking what movement to do. They also need to know how much to do and whether the recommendation is realistic for them. This is where the full-prescription setup becomes relevant. The results suggest that learned policies become more attractive when the system must make decisions under uncertainty about user tolerance and when these decisions interact over time.

A third implication concerns feedback. In many real systems, users do not provide detailed ratings after every recommendation. They do, however, accept, reject, skip, or modify what is suggested. This thesis supports the idea that such behavioral signals can be useful for personalization, even when they are weak and implicit. Skip-only feedback is therefore an interesting signal for practical recommendation systems, especially in health and fitness domains where explicit feedback is limited.

At the same time, the results also suggest that personalization based on such signals requires some caution. Skip behavior is informative, but it is not perfectly interpretable. A user may skip because the recommendation is too hard, too easy, unfamiliar, boring, painful, inconvenient, or simply badly timed. In the present thesis, skipping is mainly tied to workload, routine mismatch, and tolerance-related dynamics. That is a reasonable modeling choice, but it is still a simplification. In real-world systems, skip behavior would likely need to be combined with other signals to make personalization more reliable.

Finally, the robustness findings have a practical implication as well. Recommendation systems that adapt to users over time should be evaluated under average conditions, but also under changing conditions. The results here suggest that gradual change is not the main problem. Highly unstable or rapidly changing user behavior is harder. This points toward the importance of recalibration, uncertainty handling, and possibly more conservative adaptation strategies in real applications.

7.5. Limitations

The most important limitation of this thesis is the use of a simulator instead of real user interaction data. This follows directly from the lack of suitable public datasets for gym workout recommendation, and in that sense it is one of the main motivations for the thesis itself. Even so, it remains a limitation. The simulator makes controlled experiments and systematic comparisons possible, but it cannot fully capture the complexity of real gym behavior. Real users may skip exercises for reasons that are not modeled here, such as discomfort, pain, time pressure, equipment availability, embarrassment, boredom, or changing motivation. The current simulator captures only a subset of these factors.

A related limitation is the sim-to-real gap more broadly. The results show what PPO can learn in the designed environments, but they do not yet prove that the same gains would transfer directly to a deployed gym application. This matters especially in the skip-enabled environments, where the learned policy depends on the structure of the skip simulator and the online belief updates. If real users behave differently from these assumptions, then the learned policies may also behave differently. For that reason, this thesis should be interpreted as a strong simulation study rather than as a final validation of a deployable gym recommender.

The thesis also does not include a real user study. The original intention to collect user data was overtaken by practical delays in the ethics approval process, making it unrealistic to collect, process, and meaningfully integrate such data within the available project time. This does not invalidate the thesis, but it does limit the strength of the practical claims that can be made. The work demonstrates a promising framework and a convincing simulation-based result pattern, but not yet real-world user effectiveness.

Another limitation concerns the use of the HFRL framework as a starting point. While this thesis builds clearly on that work, the original paper does not fully specify all implementation details needed for exact reproduction. As a result, the exercise-only comparison should be understood as an informed replication and adaptation rather than a perfect line-by-line reproduction. This thesis is transparent about these uncertainties, but they still matter when interpreting the comparison.

A further limitation is that the reward design is hand-specified and based on domain-informed choices. This includes the choice of reward components, thresholds, and weights. These design choices are justified, but they still influence the behavior of the learned policy. Different reward weights or alternative

shaping strategies could produce different trade-offs. In that sense, part of the system's behavior is a consequence of how a "good" workout session is formalized in this thesis.

The prescription-quality reward is also based on proxy information rather than true personalized coaching outcomes. StrengthLevel data and the Epley-based mapping provide a practical and grounded way to model gym prescription, but they remain approximations. They do not replace direct evidence of what a specific user should optimally perform in a real gym setting. This means that the full-prescription environments are more realistic than exercise-only recommendation, but still not a complete representation of real coaching practice.

The routine structure used in the environments is another simplification. The experiments use a compact routine setup and fixed session length to keep the comparisons manageable and consistent. In practice, gym routines can vary in length, split structure, recovery demands, available equipment, and user intent. The thesis therefore studies an important and meaningful subset of gym recommendation, but not the full range of workout programming found in practice.

The evaluation scope is also limited in some respects. PPO is the central RL method used in this thesis because it is a stable and practical choice for the environments considered here. This is a reasonable choice, but it means that the thesis does not provide a broad comparison across multiple RL families. In the same way, while several baseline comparisons and robustness experiments are included, not every possible ablation or design variation could be explored within the available thesis timeframe.

Finally, the dynamic user pools should be interpreted with care. The standard dynamic pool is intentionally moderate, which supports realism, but also means that the static-versus-dynamic comparison is not a test of severe non-stationarity. The robust and chaotic pools partly address this, but they are stress tests rather than realistic settings. Together, these experiments provide a useful range of conditions, but not a complete picture of all possible forms of user change.

Despite these limitations, the thesis still makes a meaningful contribution. It provides a transparent simulator-based framework for gym workout recommendation, shows where RL does and does not provide clear benefit, and identifies the specific conditions under which learning-based methods become more convincing than simpler baselines. The limitations therefore mainly define the scope of the conclusions, rather than undermining the central findings.

8

Conclusion

This thesis studied whether reinforcement learning improves personalized workout recommendation when the problem is extended from simple exercise selection to more realistic gym-based session planning. Starting from the HFRL framework of Tragos et al., the work moved from home-fitness exercise recommendation to a gym setting in which recommendations can include both exercises and full prescription variables: sets, repetitions, and load. In addition, the thesis examined the role of skipping behavior, online personalization, and changing user characteristics over time.

The results show that the usefulness of reinforcement learning depends strongly on the structure of the recommendation problem. In the exercise-only environments, PPO consistently outperformed the weaker baselines, but it did not surpass the greedy baseline. This indicates that when the task is limited to selecting a sequence of exercises, a strong hand-crafted heuristic can still capture much of the relevant structure in the environment. In that simpler setting, reinforcement learning is effective, but not yet clearly necessary.

The picture changes in the full-prescription environments. Once the action space includes sets, repetitions, and load, PPO becomes the strongest method and outperforms all baselines, including greedy. This is the clearest result of the thesis. It shows that the advantage of reinforcement learning appears when the recommendation problem becomes richer and more realistic, and when the system must make joint decisions about both exercise selection and prescription. In that setting, the problem is no longer only about creating a plausible session structure, but also about recommending an amount of work that is appropriate for the specific user.

The thesis also shows that user interaction changes the recommendation problem itself. In the skip-enabled environments, the policy is no longer rewarded only for sessions that look good structurally, but for sessions that are actually completed by the user. PPO is able to use this interaction signal effectively and learns more adherence-aware behavior. At the same time, this comes with clear trade-offs: optimizing for adherence does not improve every reward component at once, but shifts the policy toward recommendations that are more practical under user-specific constraints. This makes skipping important as both an evaluation signal and as part of the recommendation objective.

A further contribution of the thesis is methodological. Because suitable real-world gym interaction data was not available, the work had to address the recommendation problem itself, but also how such a problem can be studied in a controlled and reproducible way. The resulting framework combines a gym-oriented environment design, synthetic user pools, full-prescription action spaces, and skip-based online personalization. In that sense, the thesis contributes a set of experimental results, as well as a concrete simulator-based setup for studying sequential gym recommendation under data scarcity.

The experiments with static, dynamic, robust, and chaotic user pools also refine the overall picture. PPO remains stable under realistic gradual user change, which suggests that the framework can handle moderate longitudinal drift without meaningful performance loss. At the same time, the robustness tests show that highly chaotic and rapidly reversing user dynamics are much harder, especially when online

personalization is involved. This indicates that the main difficulty is not simply that users change, but that user-dependent beliefs become unreliable when change is too unpredictable.

Taken together, these findings answer the main research question of the thesis. Reinforcement learning is not equally beneficial in every workout recommendation setting. Its value becomes most clear when the problem moves beyond simple exercise sequencing and toward realistic gym recommendation with full prescription, adherence effects, and user-specific interaction over time. The main contribution of this thesis is therefore twofold. First, it extends the HFRL framework to a richer gym domain with full-prescription recommendation and skip-based online personalization. Second, it shows more clearly under which conditions reinforcement learning begins to offer a convincing advantage over strong non-learning baselines.

More broadly, the thesis suggests that the practical value of RL in fitness recommendation is not in replacing simple recommenders everywhere, but in handling the parts of the problem that are genuinely sequential, personalized, and interaction-dependent. That makes the contribution of the thesis both more modest and more useful: it does not argue that RL is always the right choice, but it does show where its added complexity begins to pay off in a meaningful way.

8.1. Future Work

Although this thesis shows that reinforcement learning can be beneficial for gym recommendation in more realistic settings, several important directions remain open. The current framework already extends the HFRL framework in meaningful ways, but it still simplifies many aspects of real-world training and user behavior. Future work can therefore improve both the realism of the simulator and the scope of the recommendation problem.

8.1.1. Validation with real users

The main limitation of this thesis is that training and evaluation are performed in simulation. While the simulator is designed to reflect plausible workout behavior, it remains a model of user behavior rather than real behavior itself. An important next step would therefore be to evaluate the proposed framework with actual users.

A real-user study would make it possible to test whether the gains found in simulation also translate to practical recommendation quality, user satisfaction, and adherence in a gym setting. It would also help determine whether the skip signals and personalization mechanisms used in this thesis correspond to how real users respond to difficult, repetitive, or badly timed recommendations. Such a study could follow a similar structure to the HFRL framework, where an RL model is compared against baseline recommenders over multiple weeks.

8.1.2. Learning from real feedback signals

Related to real-user validation, future work could move beyond simulated skip behavior and make use of richer real interaction signals. In practice, users may provide many forms of implicit or explicit feedback, such as skipping an exercise, lowering the weight, ending a session early, rating a session poorly, or reporting perceived exertion.

Using these signals could improve personalization and make the system more adaptive to real-world variation in user state. It would also allow the recommendation policy to learn from feedback that is more informative than a binary skip signal alone. In this thesis, skipping is used as the main interaction signal because it is simple and directly connected to adherence, but future systems could incorporate multiple forms of feedback at once.

8.1.3. More realistic physiological modeling

The simulator in this thesis captures user drift, capacity, and workout-related skipping, but it still uses a simplified representation of fatigue, recovery, and long-term adaptation. A useful direction for future work would be to make the simulator physiologically richer.

For example, future versions could model recovery at the level of separate muscle groups in more detail,

include sleep or stress as additional context variables, or simulate longer-term strength development using more detailed progression rules. This could make the environment more realistic and also make the resulting recommendations more meaningful from a training perspective. At the same time, a more realistic simulator would make the learning problem harder and would provide a stronger test of whether RL remains beneficial.

8.1.4. Longer-term training planning

This thesis focuses on workout-session recommendation, where the agent constructs one session at a time. However, real gym programming often operates at a longer time scale. Users typically follow training plans over weeks or months, with progression, deloads, and routine variation over time.

A natural extension would therefore be to move from session-level planning to multi-session or program-level planning. In such a setting, the agent would not only decide which session is good today, but also how today's session fits into the user's longer-term training development. This could make persistence, recovery, and progression even more central to the recommendation problem and could further increase the value of reinforcement learning compared with short-horizon heuristics.

8.1.5. Richer action spaces and recommendation dimensions

The full-prescription environments in this thesis already extend the action space substantially by adding sets, repetitions, and load. Still, real gym recommendations involve more choices than these alone. Rest times, tempo, exercise substitutions, warm-up design, and progression strategy could all be incorporated into future recommendation systems.

Adding such dimensions would increase the realism of the problem and could make the gap between learning-based and rule-based approaches even more informative. At the same time, this would require careful action-space design to keep training feasible. Future work could explore hierarchical or factorized action representations to handle this increased complexity more efficiently.

8.1.6. Alternative learning methods

This thesis uses PPO as the main RL method, mainly because it is stable and practical for the environments considered here. However, PPO is only one possible approach. Future work could compare PPO to other RL methods, such as off-policy algorithms, recurrent policies for partially observable settings, or model-based methods.

This would be especially relevant in the skip-enabled environments, where hidden user factors and changing user states make the problem partly observable. Methods that maintain memory or explicitly model uncertainty may be better suited to these conditions. Comparing such methods could provide a clearer picture of which RL approaches are most suitable for personalized workout recommendation.

8.1.7. Robustness under broader forms of non-stationarity

The robustness analysis in this thesis focuses on structured drift and chaotic drift in user characteristics. This already gives useful insight into when the framework remains stable and when it begins to break down. Still, there are other forms of non-stationarity that could be studied in future work.

Examples include changes in user goals over time, temporary injuries, changes in available equipment, seasonal interruptions, or abrupt changes in training frequency. These are all realistic scenarios in which a recommendation system may need to adapt. Studying such cases would give a broader view of robustness and would help determine how well RL-based systems generalize outside the training conditions used here.

References

[1] *721 Weight Training Exercises (Personal Training Log)*. https://www.kaggle.com/datasets/joep89/weightlifting. 2024. (Visited on 04/03/2026).

[2] Manal Abdulaziz et al. “Building a Personalized Fitness Recommendation Application based on Sequential Information”. In: *International Journal of Advanced Computer Science and Applications* 12.1 (2021). DOI: 10.14569/IJACSA.2021.0120173. URL: http://dx.doi.org/10.14569/IJACSA.2021.0120173.

[3] M. Mehdi Afsar, Trafford Crump, and Behrouz H. Far. “Reinforcement Learning based Recommender Systems: A Survey”. In: *ACM Computing Surveys* 55.7 (2022), pp. 1–38. DOI: 10.1145/3543846.

[4] American College of Sports Medicine. “American College of Sports Medicine position stand. Progression models in resistance training for healthy adults”. In: *Medicine and Science in Sports and Exercise* 41.3 (2009), pp. 687–708. DOI: 10.1249/MSS.0b013e3181915670.

[5] Milad Asgari Mehrabadi et al. “PERFECT: Personalized Exercise Recommendation Framework and architECTure”. In: *ACM Trans. Comput. Healthcare* 5.4 (Nov. 2024). DOI: 10.1145/3696425. URL: https://doi.org/10.1145/3696425.

[6] Jakim Berndsen, Barry Smyth, and Aonghus Lawlor. “A Collaborative Filtering Approach to Successfully Completing The Marathon”. In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2020, pp. 653–658. DOI: 10.1109/ICMLA51294.2020.00108.

[7] Jakim Berndsen, Barry Smyth, and Aonghus Lawlor. “Fit to Run: Personalised Recommendations for Marathon Training”. In: *Proceedings of the 14th ACM Conference on Recommender Systems*. RecSys ’20. Virtual Event, Brazil: Association for Computing Machinery, 2020, pp. 480–485. ISBN: 9781450375832. DOI: 10.1145/3383313.3412228. URL: https://doi.org/10.1145/3383313.3412228.

[8] Jakim Berndsen, Barry Smyth, and Aonghus Lawlor. “Pace my race: recommendations for marathon running”. In: *Proceedings of the 13th ACM Conference on Recommender Systems*. RecSys ’19. Copenhagen, Denmark: Association for Computing Machinery, 2019, pp. 246–250. ISBN: 9781450362436. DOI: 10.1145/3298689.3346991. URL: https://doi.org/10.1145/3298689.3346991.

[9] Ine Coppens, Toon De Pessemier, and Luc Martens. “Balancing Habit Repetition and New Activity Exploration: A Longitudinal Micro-Randomized Trial in Physical Activity Recommendations”. In: *Proceedings of the 18th ACM Conference on Recommender Systems*. RecSys ’24. Bari, Italy: Association for Computing Machinery, 2024, pp. 1147–1151. ISBN: 9798400705052. DOI: 10.1145/3640457.3691715. URL: https://doi.org/10.1145/3640457.3691715.

[10] Ine Coppens, Toon De Pessemier, and Luc Martens. “Repeating my Workouts or Exploring new Activities? A Longitudinal Micro-Randomized User Study for Physical Activity Recommender Systems”. In: *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*. UMAP Adjunct ’24. Cagliari, Italy: Association for Computing Machinery, 2024, pp. 176–182. ISBN: 9798400704666. DOI: 10.1145/3631700.3664867. URL: https://doi.org/10.1145/3631700.3664867.

[11] Ine Coppens, Toon De Pessemier, and Luc Martens. “Explaining Decision-Making between Exploration and Repetition: Key Factors for Physical Activity Recommendations”. In: *HealthRecSys 2024: Proceedings of the 6th International Workshop on Health Recommender Systems co-located with RecSys 2024*. Ed. by Hanna Hauptman, Christoph Trattner, and Helma Torkamaan. Vol. 3823. CEUR Workshop Proceedings. Workshop date: October 18, 2024. Bari, Italy: CEUR-WS.org, Nov. 2024, pp. 8–14. URL: https://ceur-ws.org/Vol-3823/2_coppens_explaining__168.pdf.

[12] Ine Coppens, Toon De Pessemier, and Luc Martens. “Investigating different recommender algorithms in the domain of physical activity recommendations: a longitudinal between-subjects user study”. In: *User Modeling and User-Adapted Interaction* 35.1 (Feb. 2025), p. 6. ISSN: 1573-1391. DOI: 10.1007/s11257-025-09427-3. URL: https://doi.org/10.1007/s11257-025-09427-3.

[13] Saumil Dharia et al. “Social recommendations for personalized fitness assistance”. In: *Personal and Ubiquitous Computing* 22.2 (Apr. 2018), pp. 245–257. ISSN: 1617-4917. DOI: 10.1007/s00779-017-1039-8. URL: https://doi.org/10.1007/s00779-017-1039-8.

[14] Gaetano Dibenedetto et al. “Fine-Tuning Large Multimodal Models for Fitness Action Quality Assessment”. In: *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*. UMAP Adjunct ’25. Association for Computing Machinery, 2025, pp. 39–44. ISBN: 9798400713996. DOI: 10.1145/3708319.3733684. URL: https://doi.org/10.1145/3708319.3733684.

[15] Gaetano Dibenedetto et al. “Lift It Up Right: A Recommender System for Safer Lifting Postures”. In: *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. RecSys ’25. Association for Computing Machinery, 2025, pp. 1222–1227. ISBN: 9798400713644. DOI: 10.1145/3705328.3759314. URL: https://doi.org/10.1145/3705328.3759314.

[16] Boyd Epley. “Poundage Chart”. In: *Boyd Epley Workout*. Lincoln, NE: Body Enterprises, 1985, p. 86.

[17] *Exercise Recognition from Wearable Sensors*. https://github.com/microsoft/Exercise-Recognition-from-Wearable-Sensors?tab=readme-ov-file. 2020. (Visited on 04/03/2026).

[18] *ExerciseDB V1*. https://github.com/ExerciseDB/exercisedb-api. 2025. (Visited on 09/04/2025).

[19] Gunther Eysenbach. “The law of attrition”. In: *Journal of Medical Internet Research* 7.1 (2005), e11. DOI: 10.2196/jmir.7.1.e11.

[20] Ciara Feely et al. “A Case-Based Reasoning Approach to Post-injury Training Recommendations for Marathon Runners”. In: *Case-Based Reasoning Research and Development: 32nd International Conference, ICCBR 2024, Merida, Mexico, July 1–4, 2024, Proceedings*. Merida, Mexico: Springer-Verlag, 2024, pp. 338–353. ISBN: 978-3-031-63645-5. DOI: 10.1007/978-3-031-63646-2_22. URL: https://doi.org/10.1007/978-3-031-63646-2_22.

[21] Ciara Feely et al. “A Case-Based Reasoning Approach to Predicting and Explaining Running Related Injuries”. In: *Case-Based Reasoning Research and Development: 29th International Conference, ICCBR 2021, Salamanca, Spain, September 13–16, 2021, Proceedings*. Salamanca, Spain: Springer-Verlag, 2021, pp. 79–93. ISBN: 978-3-030-86956-4. DOI: 10.1007/978-3-030-86957-1_6. URL: https://doi.org/10.1007/978-3-030-86957-1_6.

[22] Ciara Feely et al. “An Extended Case-Based Approach to Race-Time Prediction for Recreational Marathon Runners”. In: *Case-Based Reasoning Research and Development: 30th International Conference, ICCBR 2022, Nancy, France, September 12–15, 2022, Proceedings*. Nancy, France: Springer-Verlag, 2022, pp. 335–349. ISBN: 978-3-031-14922-1. DOI: 10.1007/978-3-031-14923-8_22. URL: https://doi.org/10.1007/978-3-031-14923-8_22.

[23] Ciara Feely et al. “Modelling the Training Practices of Recreational Marathon Runners to Make Personalised Training Recommendations”. In: *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. UMAP ’23. Limassol, Cyprus: Association for Computing Machinery, 2023, pp. 183–193. ISBN: 9781450399326. DOI: 10.1145/3565472.3592952. URL: https://doi.org/10.1145/3565472.3592952.

[24] Ciara Feely et al. “Recommending Personalised Targeted Training Adjustments for Marathon Runners”. In: *Proceedings of the 18th ACM Conference on Recommender Systems*. RecSys ’24. Bari, Italy: Association for Computing Machinery, 2024, pp. 1051–1056. ISBN: 9798400705052. DOI: 10.1145/3640457.3688192. URL: https://doi.org/10.1145/3640457.3688192.

[25] Ciara Feely et al. “Using Case-Based Reasoning to Predict Marathon Performance and Recommend Tailored Training Plans”. In: *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings*. Salamanca, Spain: Springer-Verlag, 2020, pp. 67–81. ISBN: 978-3-030-58341-5. DOI: 10.1007/978-3-030-58342-2_5. URL: https://doi.org/10.1007/978-3-030-58342-2_5.

[26] Ciara Feely et al. “Using Pseudo Cases and Stratified Case-Based Reasoning to Generate and Evaluate Training Adjustments for Marathon Runners”. In: *Artificial Intelligence XLI: 44th SGAI International Conference on Artificial Intelligence, AI 2024, Cambridge, UK, December 17–19, 2024, Proceedings, Part II*. Cambridge, United Kingdom: Springer-Verlag, 2024, pp. 88–101. ISBN: 978-3-031-77917-6. DOI: 10.1007/978-3-031-77918-3_7. URL: https://doi.org/10.1007/978-3-031-77918-3_7.

[27] Mihai Fieraru et al. “AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9914–9923. DOI: 10.1109/CVPR46437.2021.00979.

[28] *Fit3D Dataset*. https://fit3d.imar.ro/. 2021. (Visited on 04/03/2026).

[29] *Fitness Tracker Dataset*. https://www.kaggle.com/datasets/nadeemajeedch/fitness-tracker-dataset/data. 2025. (Visited on 04/03/2026).

[30] *Fitness-AQA*. https://github.com/ParitoshParmar/Fitness-AQA. 2023. (Visited on 04/03/2026).

[31] *FitRec Dataset*. https://cseweb.ucsd.edu/~jmcauley/datasets/fitrec.html. 2019. (Visited on 04/03/2026).

[32] *Free Exercise DB*. https://github.com/yuhonas/free-exercise-db. 2025. (Visited on 04/03/2026).

[33] *Functional Fitness Exercise Database*. https://strengthtoovercome.com/functional-fitness-exercise-database. 2025. (Visited on 04/03/2026).

[34] Fabio Gasparetti, Luca Maria Aiello, and Daniele Quercia. “Evaluating the efficacy of traditional fitness tracker recommendations”. In: *Companion Proceedings of the 24th International Conference on Intelligent User Interfaces*. IUI ’19 Companion. Marina del Ray, California: Association for Computing Machinery, 2019, pp. 15–16. ISBN: 9781450366731. DOI: 10.1145/3308557.3308716. URL: https://doi.org/10.1145/3308557.3308716.

[35] Xiaonan Guo, Jian Liu, and Yingying Chen. “FitCoach: Virtual fitness coach empowered by wearable mobile devices”. In: *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 2017, pp. 1–9. DOI: 10.1109/INFOCOM.2017.8057208.

[36] *Gym Exercise Dataset*. https://www.kaggle.com/datasets/niharika41298/gym-exercise-data. 2025. (Visited on 04/03/2026).

[37] *Gym Exercises (Kaggle catalogue)*. https://www.kaggle.com/datasets/willianoliveiragibin/gym-exercises. 2025. (Visited on 04/03/2026).

[38] *Gym Exercises Dataset (Gigasheet)*. https://www.gigasheet.com/sample-data/gym-exercises-dataset---sheet1. 2025. (Visited on 04/03/2026).

[39] *Gym Workout IMU Dataset*. https://www.kaggle.com/datasets/shakthisairam123/gym-workout-imu-dataset. 2023. (Visited on 04/03/2026).

[40] Qian He et al. “RecFit: a context-aware system for recommending physical activities”. In: *Proceedings of the 1st Workshop on Mobile Medical Applications*. MMA ’14. Memphis, Tennessee: Association for Computing Machinery, 2014, pp. 34–39. ISBN: 9781450331906. DOI: 10.1145/2676431.2676439. URL: https://doi.org/10.1145/2676431.2676439.

[41] Eugene Ie et al. “RecSim: A Configurable Simulation Platform for Recommender Systems”. In: *arXiv preprint arXiv:1909.04847* (2019). DOI: 10.48550/arXiv.1909.04847.

[42] Abhishek Jaiswal, Gautam Chauhan, and Nisheeth Srivastava. “Using Learnable Physics for Real-Time Exercise Form Recommendations”. In: *Proceedings of the 17th ACM Conference on Recommender Systems*. RecSys ’23. Singapore, Singapore: Association for Computing Machinery, 2023, pp. 688–695. ISBN: 9798400702419. DOI: 10.1145/3604915.3608816. URL: https://doi.org/10.1145/3604915.3608816.

[43] Nesrine Kadri, Ameni Ellouze, and Mohamed Ksantini. “Recommendation system for human physical activities using smartphones”. In: *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. 2020, pp. 1–4. DOI: 10.1109/ICCIS49240.2020.9257671.

[44] Dale A. LeSuer et al. “The Accuracy of Prediction Equations for Estimating 1-RM Performance in the Bench Press, Squat, and Deadlift”. In: *Journal of Strength and Conditioning Research* 11.4 (1997), pp. 211–213. DOI: 10.1519/00124278-199711000-00001.

[45] Xiao Liu et al. “Privacy-Preserving Personalized Fitness Recommender System P3FitRec: A Multi-level Deep Learning Approach”. In: *ACM Trans. Knowl. Discov. Data* 17.6 (Apr. 2023). ISSN: 1556-4681. DOI: 10.1145/3572899. URL: https://doi.org/10.1145/3572899.

[46] Arash Mahyari and Peter Pirolli. “Physical Exercise Recommendation and Success Prediction Using Interconnected Recurrent Neural Networks”. In: *2021 IEEE International Conference on Digital Health (ICDH)*. 2021, pp. 148–153. DOI: 10.1109/ICDH52753.2021.00027.

[47] Cathal McConnell and Barry Smyth. “Going Further with Cases: Using Case-Based Reasoning to Recommend Pacing Strategies for Ultra-Marathon Runners”. In: *Case-Based Reasoning Research and Development: 27th International Conference, ICCBR 2019, Otzenhausen, Germany, September 8–12, 2019, Proceedings*. Otzenhausen, Germany: Springer-Verlag, 2019, pp. 358–372. ISBN: 978-3-030-29248-5. DOI: 10.1007/978-3-030-29249-2_24. URL: https://doi.org/10.1007/978-3-030-29249-2_24.

[48] *MM-Fit Dataset*. https://mmfit.github.io/. 2021. (Visited on 04/03/2026).

[49] Heleen Muijlwijk et al. “Benefits of Human-AI Interaction for Expert Users Interacting with Prediction Models: a Study on Marathon Running”. In: *Proceedings of the 29th International Conference on Intelligent User Interfaces*. IUI ’24. Greenville, SC, USA: Association for Computing Machinery, 2024, pp. 245–258. ISBN: 9798400705083. DOI: 10.1145/3640543.3645205. URL: https://doi.org/10.1145/3640543.3645205.

[50] *Muscle & Strength Exercise Video Database*. https://www.muscleandstrength.com/exercises. 2025. (Visited on 04/03/2026).

[51] *Muscle & Strength Workout Routines*. https://www.muscleandstrength.com/workout-routines. 2025. (Visited on 04/03/2026).

[52] *MuscleWiki (clone repository)*. https://github.com/AlimKhan76/musclewiki. 2025. (Visited on 04/03/2026).

[53] *MuscleWiki Exercise Directory*. https://musclewiki.com/directory. 2025. (Visited on 04/03/2026).

[54] Jianmo Ni, Larry Muhlstein, and Julian McAuley. “Modeling Heart Rate and Activity Data for Personalized Fitness Recommendation”. In: *The World Wide Web Conference*. WWW ’19. San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 1343–1353. ISBN: 9781450366748. DOI: 10.1145/3308558.3313643. URL: https://doi.org/10.1145/3308558.3313643.

[55] Paolo Pilloni et al. “Recommendation in Persuasive eHealth Systems: an Effective Strategy to Spot Users’ Losing Motivation to Exercise”. In: *Proceedings of the 2nd International Workshop on Health Recommender Systems co-located with the 11th International Conference on Recommender Systems (RecSys 2017), Como, Italy, August 31, 2017*. Ed. by David Elsweiler et al. Vol. 1953. CEUR Workshop Proceedings. CEUR-WS.org, 2017, pp. 6–9. URL: https://ceur-ws.org/Vol-1953/healthRecSys17%5C_paper%5C_5.pdf.

[56] *RecGym: Gym Workouts Data Set*. https://www.kaggle.com/datasets/zhaxidelebsz/10-gym-exercises-with-615-abstracted-features. 2025. (Visited on 04/03/2026).

[57] *RepCount Dataset*. https://svip-lab.github.io/dataset/RepCount_dataset.html. 2023. (Visited on 04/03/2026).

[58] Rijksinstituut voor Volksgezondheid en Milieu (RIVM). *Gezond gewicht*. Leefstijlmonitor page with BMI category definitions and Dutch prevalence figures; page updated 2026-03-13. 2026. URL: https://www.rivm.nl/leefstijlmonitor/gezond-gewicht (visited on 04/03/2026).

[59] Caroline T.M. van Rossum et al. *The Diet of the Dutch: Results of the Dutch National Food Consumption Survey 2019–2021 on Food Consumption and Evaluation with Dietary Guidelines*. Tech. rep. RIVM report 2022-0190. National Institute for Public Health and the Environment (RIVM), 2023. DOI: 10.21945/RIVM-2022-0190. URL: https://www.rivm.nl/bibliotheek/rapporten/2022-0190.pdf (visited on 04/03/2026).

[60] Alan Said. *Early Explorations of Recommender Systems for Physical Activity and Well-being*. 2025. arXiv: 2508.07980 [cs.HC]. URL: https://arxiv.org/abs/2508.07980.

[61] Odnan Ref Sanchez et al. “A recommendation approach for user privacy preferences in the fitness domain”. In: *User Modeling and User-Adapted Interaction* 30.3 (July 2020), pp. 513–565. ISSN: 1573-1391. DOI: 10.1007/s11257-019-09246-3. URL: https://doi.org/10.1007/s11257-019-09246-3.

[62] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017). DOI: 10.48550/arXiv.1707.06347.

[63] *Sensor-Based Gym Physical Exercise Recognition: Data Acquisition and Experiments (data link)*. https://doi.org/10.3390/s22072489. 2022. (Visited on 04/03/2026).

[64] Barry Smyth. “Recommender Systems: A Healthy Obsession”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 9790–9794. DOI: 10.1609/aaai.v33i01.33019790. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5052.

[65] Barry Smyth. “Running Recommendations: Personalisation Opportunities for Health and Fitness”. In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. UMAP ’18. Singapore, Singapore: Association for Computing Machinery, 2018, p. 1. ISBN: 9781450355896. DOI: 10.1145/3209219.3209269. URL: https://doi.org/10.1145/3209219.3209269.

[66] Barry Smyth and Pádraig Cunningham. “A Novel Recommender System for Helping Marathoners to Achieve a New Personal-Best”. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys ’17. Como, Italy: Association for Computing Machinery, 2017, pp. 116–120. ISBN: 9781450346528. DOI: 10.1145/3109859.3109874. URL: https://doi.org/10.1145/3109859.3109874.

[67] Barry Smyth and Pádraig Cunningham. “Marathon race planning: a case-based reasoning approach”. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI’18. Stockholm, Sweden: AAAI Press, 2018, pp. 5364–5368. ISBN: 9780999241127.

[68] Sport en Bewegen in Cijfers. *Sportdeelname wekelijks*. CBS/RIVM-based Dutch weekly sports participation indicator. 2025. URL: https://www.sportenbewegenincijfers.nl/kernindicatoren/sportdeelname-wekelijks (visited on 04/03/2026).

[69] *Starting Strength Training Logs (Forum)*. https://startingstrength.com/resources/forum/forum152/. 2025. (Visited on 04/03/2026).

[70] Statistics Netherlands (CBS). *Leisure - Figures - Society | Trends in the Netherlands 2018*. Includes figures on fitness-centre membership in the Netherlands. 2018. URL: https://longreads.cbs.nl/trends18-eng/society/figures/leisure (visited on 04/03/2026).

[71] Strength Level. *Strength Level Calculator and Strength Standards*. https://strengthlevel.com/strength-standards. 2026. (Visited on 05/14/2026).

[72] *StrengthLog Exercise Directory*. https://www.strengthlog.com/exercise-directory/. 2025. (Visited on 04/03/2026).

[73] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. The MIT Press, 2018. ISBN: 9780262039246.

[74] Elias Tragos et al. “Keeping People Active and Healthy at Home Using a Reinforcement Learning-based Fitness Recommendation Framework”. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. Ed. by Edith Elkind. AI for Good. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 6237–6245. DOI: 10.24963/ijcai.2023/692. URL: https://doi.org/10.24963/ijcai.2023/692.

[75] Tin Trung Tran et al. “Recommender System with Artificial Intelligence for Fitness Assistance System”. In: *2018 15th International Conference on Ubiquitous Robots (UR)*. 2018, pp. 489–492. DOI: 10.1109/URAI.2018.8441895.

[76] Chelsea C. White and Douglas J. White. “Markov decision processes”. In: *European Journal of Operational Research* 39.1 (1989), pp. 1–16. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(89)90348-2. URL: https://www.sciencedirect.com/science/article/pii/0377221789903482.

[77] *Workout Preferences and Fitness Goals Dataset*. https://www.kaggle.com/datasets/sumedh1507/fitness-and-workout-dataset. 2024. (Visited on 04/03/2026).

[78] World Health Organization. *WHO Guidelines on Physical Activity and Sedentary Behaviour*. Geneva, Switzerland: World Health Organization, 2020. URL: https://iris.who.int/handle/10665/336656 (visited on 04/02/2026).

[79] *Wrkout Exercises (JSON catalogue)*. https://github.com/wrkout/exercises.json. 2025. (Visited on 04/03/2026).

[80] Jie Zheng and Jing Yang. “Research on the Design of Personalized Exercise Recommendation System Based on Health Management of Middle-Aged and Elderly People”. In: *Design, User Experience, and Usability*. Ed. by Martin Schrepp. Cham: Springer Nature Switzerland, 2025, pp. 307–323. ISBN: 978-3-031-93224-3.

A

Literature Search Strategy

This appendix provides the database-specific search strings used to identify literature for the related-work chapter. They are included for transparency and reproducibility.

A.1. Exclusion criteria

Papers were excluded when the retrieved terms were used with a different meaning than intended in this thesis. In practice, this mainly affected the terms “exercise”, “running”, “fitness”, and “gym”, which are frequently used in artificial-intelligence literature in unrelated ways, such as coding exercises, running algorithms, fitness functions, or Gym-style simulation environments. Papers were also excluded when they were not about AI-supported recommendation, personalization, or feedback in the domain of physical activity, fitness, or exercise.

Table A.1: Database-specific search strings used in the literature search.

Database	Search string
Scopus	`TITLE(((recommend* OR "recommender system*" OR "recommendation system*" OR recommender) AND ("physical activit*" OR exercising OR fitness OR workout* OR marathon OR lifting OR "action quality" OR "exercise form" OR "posture")) OR "Recommender System: A Healthy Obsession" OR ("collaborative filtering" AND (marathon OR running)) OR ("fitness coach" AND wearable*) OR (AI* AND "fitness training")) AND ABS("machine learning" OR "deep learning" OR "artificial intelligence" OR "reinforcement learning" OR bandit OR "collaborative filtering" OR "content-based" OR hybrid OR "context-aware" OR "computer vision" OR pose OR skeleton OR "action quality") AND ABS(wearable* OR sensor* OR "heart rate" OR acceleromet* OR smartwatch* OR "fitness tracker*" OR "step goal*" OR "training plan*" OR "workout plan*" OR coach OR "virtual coach" OR "personalized feedback" OR "fitness" OR "recommender system") AND NOT ABS(diet OR recipe OR sleep OR "mental health" OR depress* OR anxi* OR stress OR smoke OR "smoking cessation" OR rehab* OR clinical OR patient* OR oncology OR diabetes OR music) AND (PUBYEAR > 2009 AND PUBYEAR < 2026) AND (LIMIT-TO (LANGUAGE, "English"))`
Web of Science	`TI=(((recommend* OR "recommender system*" OR "recommendation system*" OR recommender) NEAR/5 ("physical activit*" OR exercising OR fitness OR workout* OR marathon OR lifting OR "action quality" OR "exercise form" OR "posture")) OR "Recommender System: A Healthy Obsession" OR ("collaborative filtering" NEAR/3 (marathon OR running)) OR ("fitness coach" NEAR/3 wearable*) OR (("artificial intelligence" NEAR/3 "fitness training") OR (AI NEAR/3 "fitness training"))) AND TS=("machine learning" OR "deep learning" OR "artificial intelligence" OR "reinforcement learning" OR bandit OR "collaborative filtering" OR "content-based" OR hybrid OR "context-aware" OR "computer vision" OR pose OR skeleton OR "action quality") AND TS=(wearable* OR sensor* OR "heart rate" OR acceleromet* OR smartwatch* OR "fitness tracker*" OR "step goal*" OR "training plan*" OR "workout plan*" OR coach OR "virtual coach" OR "personalized feedback" OR fitness OR "recommender system") NOT TS=(diet OR recipe OR sleep OR "mental health" OR depress* OR anxi* OR stress OR smoke OR "smoking cessation" OR rehab* OR clinical OR patient* OR oncology OR diabetes OR music)`
ACM Digital Library	`((Title:("recommender system" OR "recommendation system" OR "recommender" OR "recommendation") OR Abstract:("recommender system" OR "recommendation system" OR "recommender" OR "recommendation")) AND (Title:("physical activity" OR "physical activities" OR "exercising" OR "marathon" OR "lifting" OR "fitness" OR "workout" OR "workouts") OR Abstract:("physical activity" OR "physical activities" OR "exercising" OR "marathon" OR "lifting" OR "fitness" OR "workout" OR "workouts")) AND ((Title:("exercise form" OR "posture" OR "pace" OR "gym" OR "sports" OR "running" OR "strength training" OR "tracker") OR Abstract:("exercise form" OR "posture" OR "pace" OR "gym" OR "sports" OR "running" OR "strength training" OR "tracker")) OR (Title:("collaborative filtering" OR "computer vision" OR "content-based" OR "knowledge-based" OR "reinforcement learning" OR "physics") OR Abstract:("collaborative filtering" OR "computer vision" OR "content-based" OR "knowledge-based" OR "reinforcement learning" OR "physics"))) AND NOT (Title:("employee training" OR "photogenic" OR "POIs" OR "Budgeted Embedding Table" OR "social heritage" OR "fuzzy inference systems") OR Abstract:("employee training" OR "photogenic" OR "POIs" OR "Budgeted Embedding Table" OR "social heritage" OR "fuzzy inference systems")))`

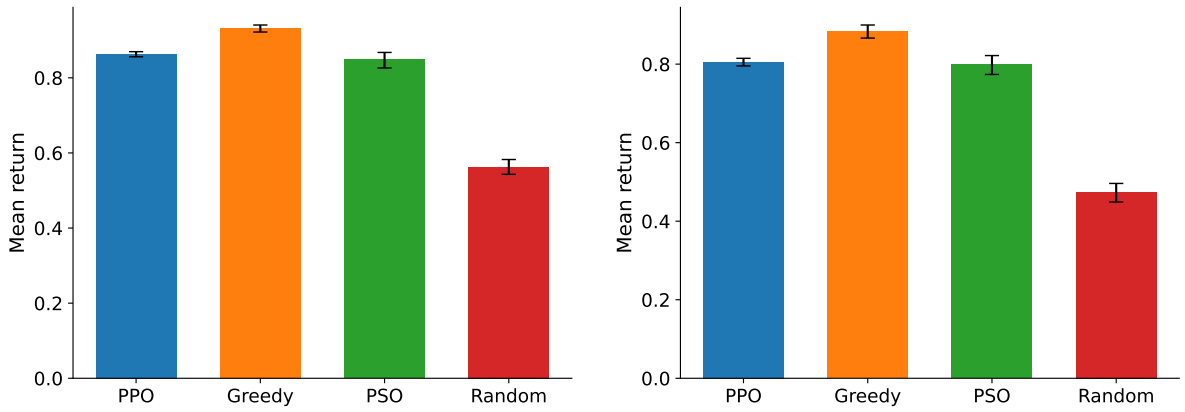
B

Additional Results Figures

This appendix contains additional result figures that are omitted from Chapter 6 for brevity. In the main text, representative figures are shown when they are sufficient to explain the result qualitatively. The figures below provide the corresponding plots for the remaining environments and pool settings.

B.1. Baseline comparisons: dynamic-pool figures

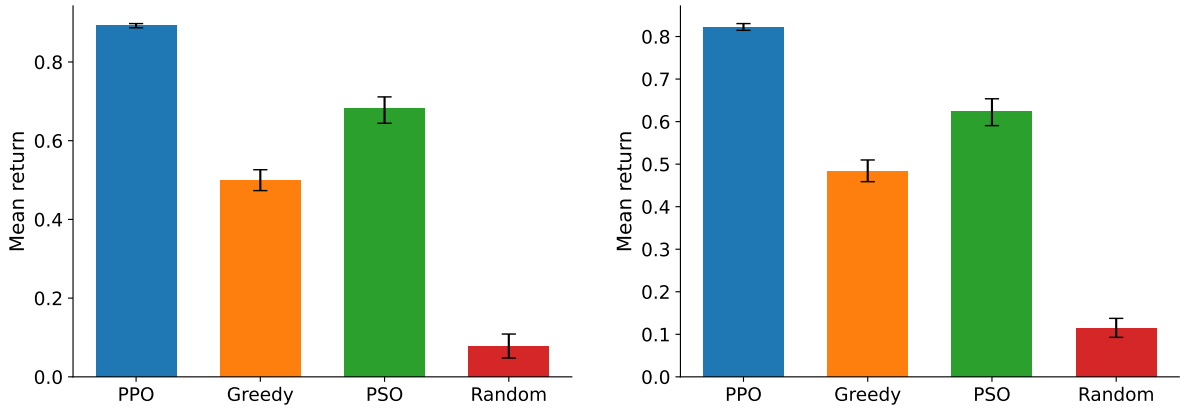
Figure B.1 provides the dynamic-pool baseline comparisons corresponding to Figure 6.1 in Section 6.1. Figure B.2 provides the corresponding dynamic-pool comparisons for the full-prescription environments, complementing Figure 6.3.



(a) ExerciseOnlyNoSkip, dynamic pool

(b) ExerciseOnlySkip, dynamic pool

Figure B.1: Dynamic-pool baseline comparisons in the exercise-only setting using mean return $\pm$ 95% bootstrap confidence intervals. These are the dynamic-pool counterparts of Figure 6.1.



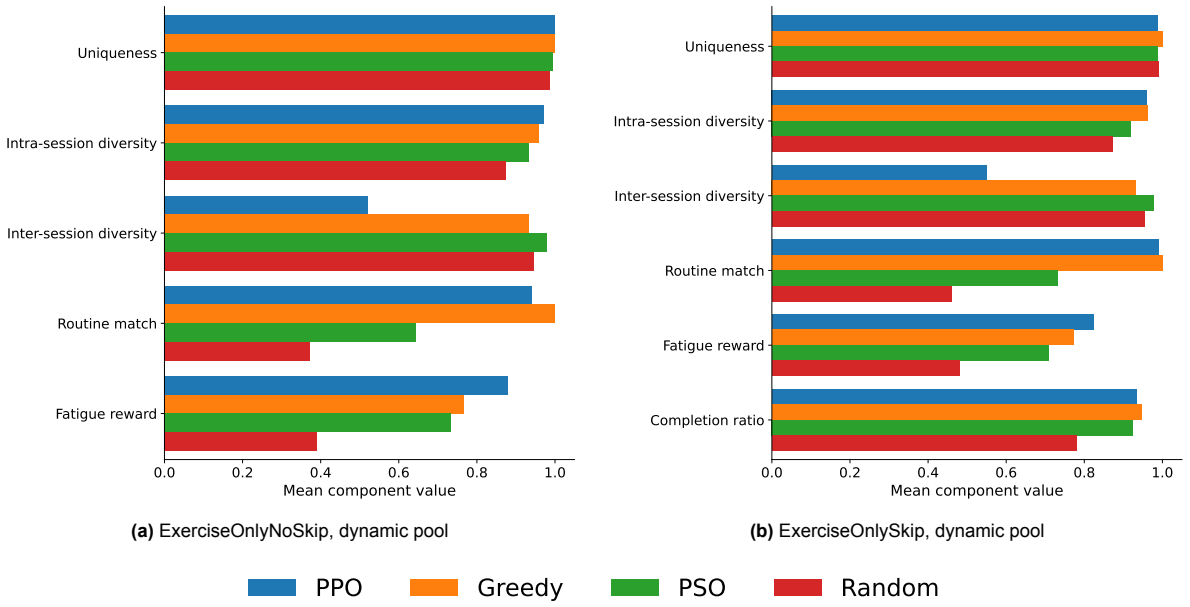
(a) FullPrescriptionNoSkip, dynamic pool

(b) FullPrescriptionSkip, dynamic pool

Figure B.2: Dynamic-pool baseline comparisons in the full-prescription setting using mean return ± 95% bootstrap confidence intervals. These are the dynamic-pool counterparts of Figure 6.3.

B.2. Baseline comparisons: dynamic-pool component means

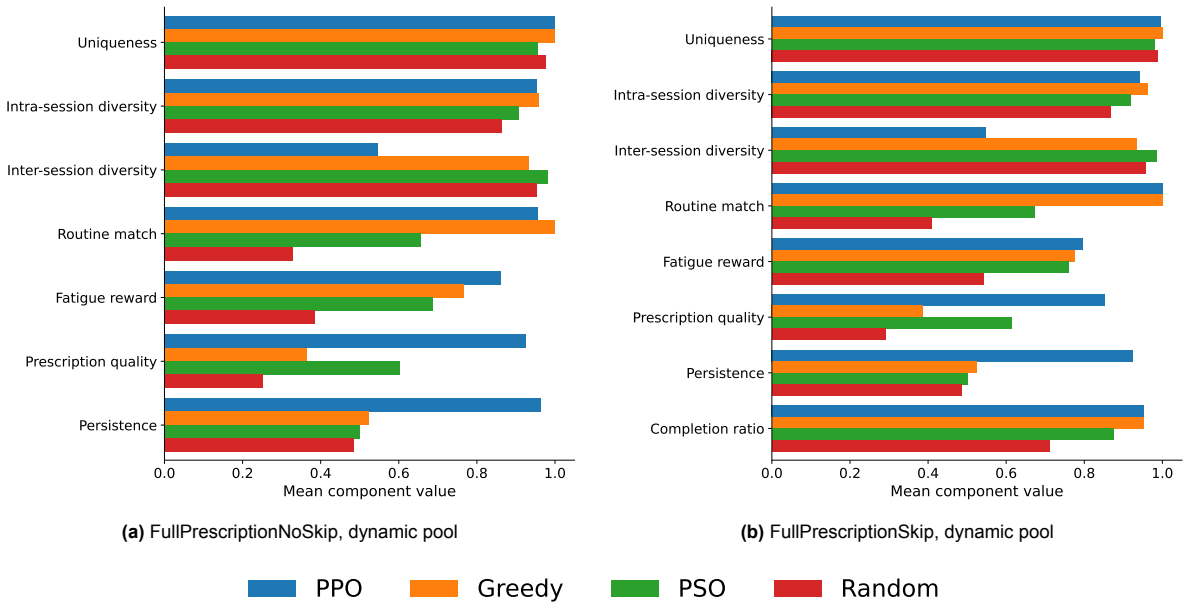
Figures B.3 and B.4 provide the dynamic-pool reward-component means corresponding to Figures 6.2 and 6.4 in the main text.



(a) ExerciseOnlyNoSkip, dynamic pool

(b) ExerciseOnlySkip, dynamic pool

Figure B.3: Dynamic-pool reward-component means in the exercise-only setting. A shared method legend is shown below the plots. These complement Figure 6.2.



(a) FullPrescriptionNoSkip, dynamic pool

(b) FullPrescriptionSkip, dynamic pool

Figure B.4: Dynamic-pool reward-component means in the full-prescription setting. A shared method legend is shown below the plots. These complement Figure 6.4.

B.3. Additional PPO training curves for static and dynamic pools

Figure B.5 provides the PPO training curves omitted from Figure 6.7 in Chapter 6. The main text shows representative curves for `ExerciseOnlyNoSkip` and `FullPrescriptionSkip`; the figure below adds the remaining environments.

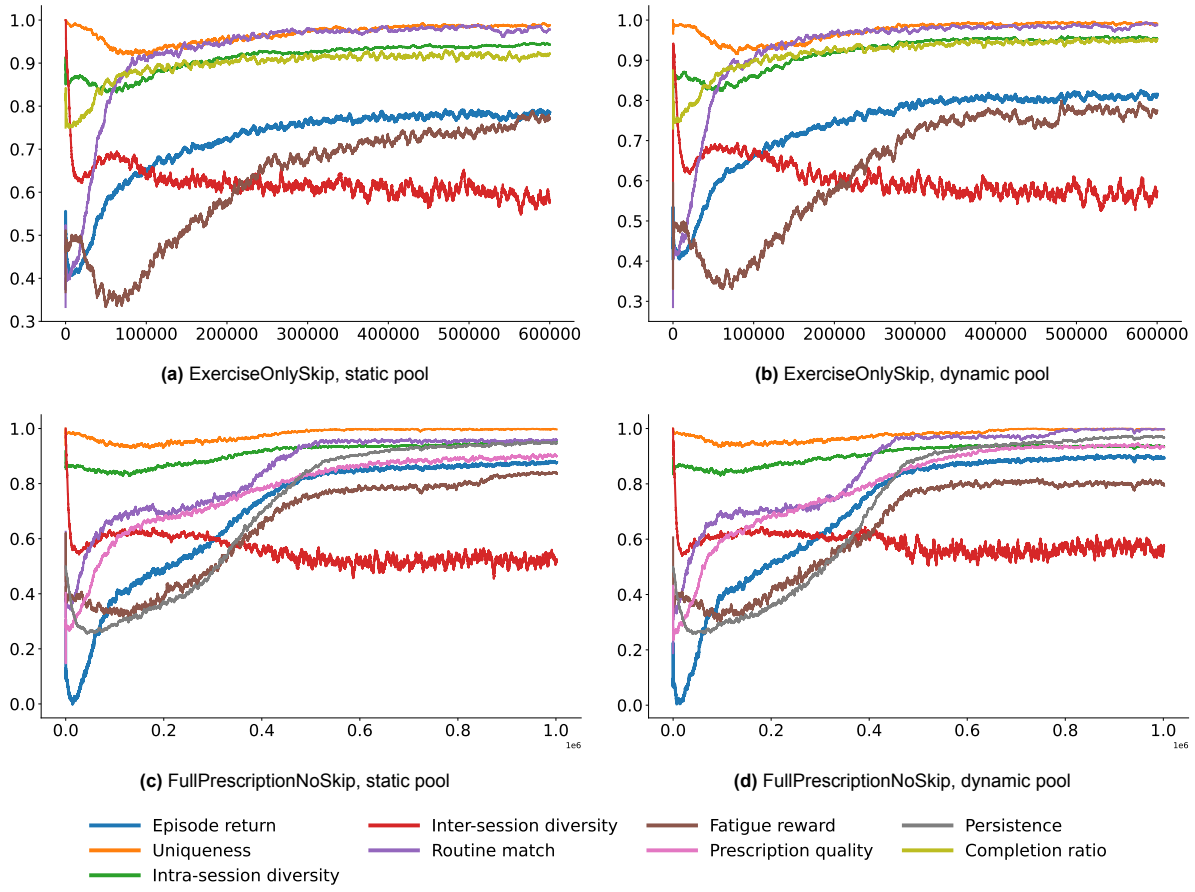
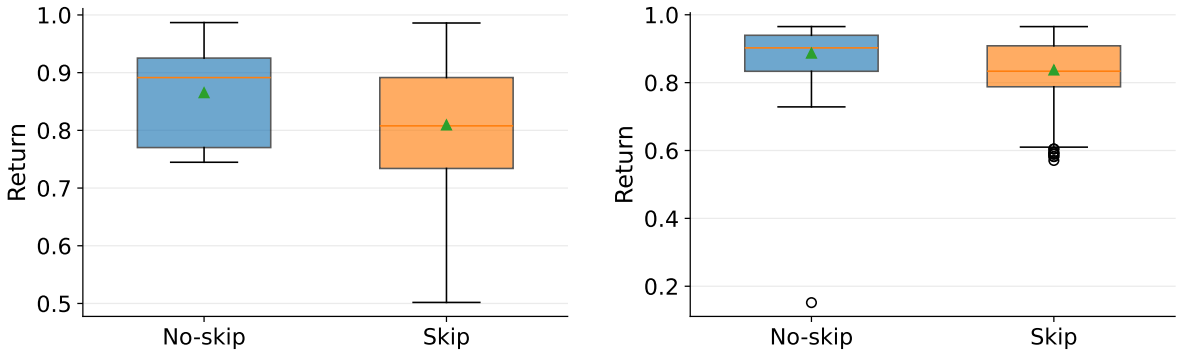


Figure B.5: Additional PPO training curves for the static versus dynamic comparison. Shared color legends are shown below each row. These are the environments not shown in Figure 6.7.

B.4. No-skip versus skip: dynamic-pool figures

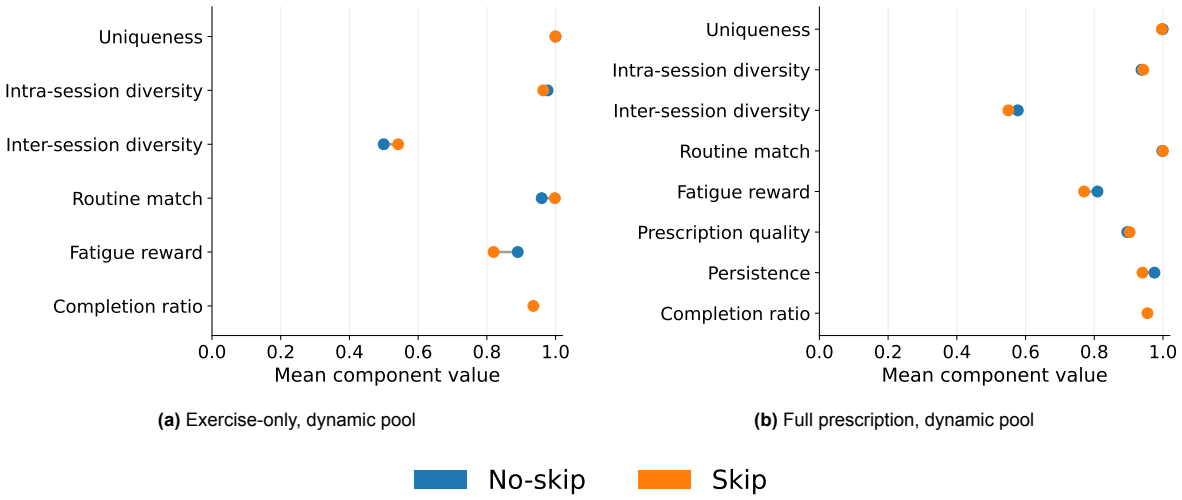
Figures B.6 and B.7 provide the dynamic-pool counterparts of Figures 6.8 and 6.9. Figure B.8 provides the dynamic-pool PPO training curves corresponding to Figure 6.10.



(a) Exercise-only, dynamic pool

(b) Full prescription, dynamic pool

Figure B.6: Return comparison between no-skip and skip-enabled PPO environments for the dynamic pool. These are the dynamic-pool counterparts of Figure 6.8.



(a) Exercise-only, dynamic pool

(b) Full prescription, dynamic pool

Figure B.7: Component means for no-skip and skip-enabled PPO environments for the dynamic pool. A shared color legend is shown below the figure. These are the dynamic-pool counterparts of Figure 6.9.

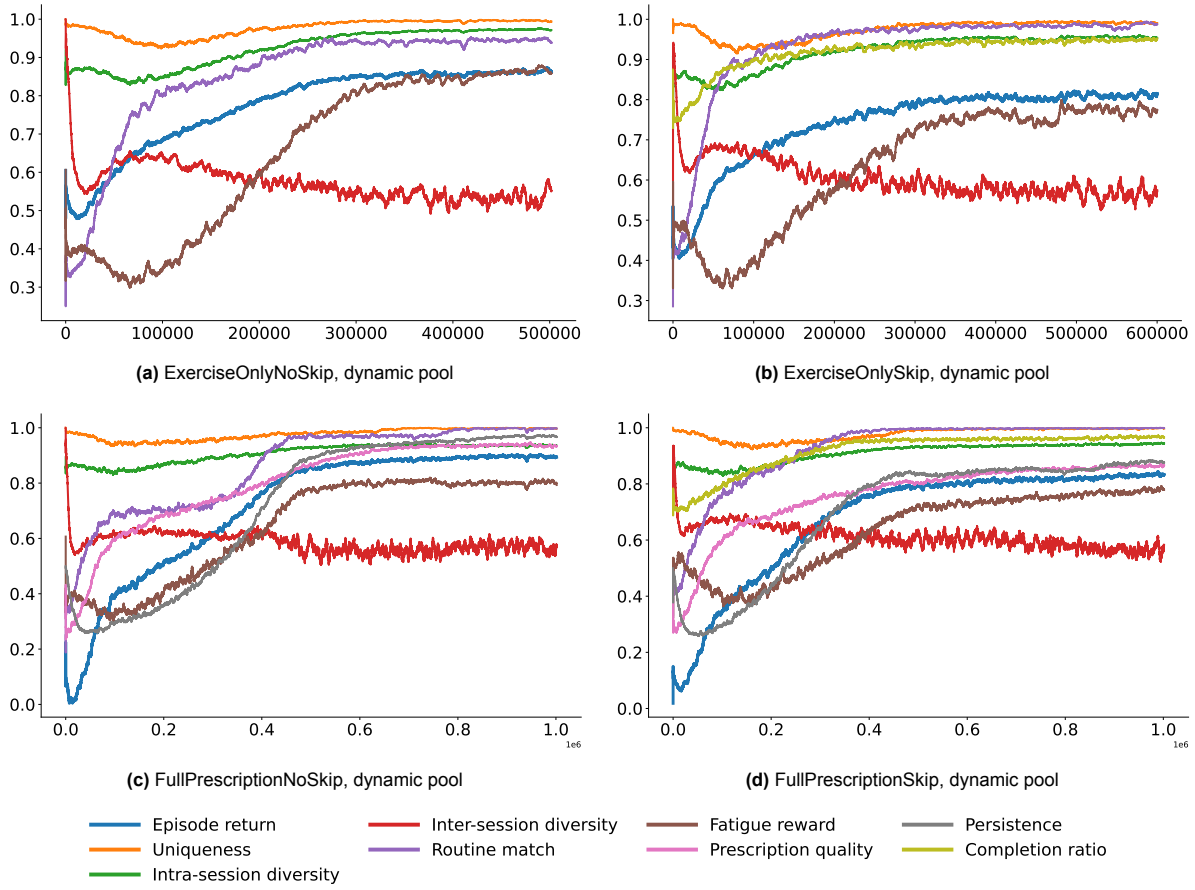


Figure B.8: Dynamic-pool PPO training curves comparing no-skip and skip-enabled environments. Shared color legends are shown below each row. These complement Figure 6.10, which shows the static-pool versions.

B.5. Additional robustness training curves for the exercise-only environments

Figure B.9 provides the exercise-only robustness training curves omitted from Figure 6.13. The main text shows representative robustness curves for the full-prescription environments; the figure below adds the exercise-only counterparts.

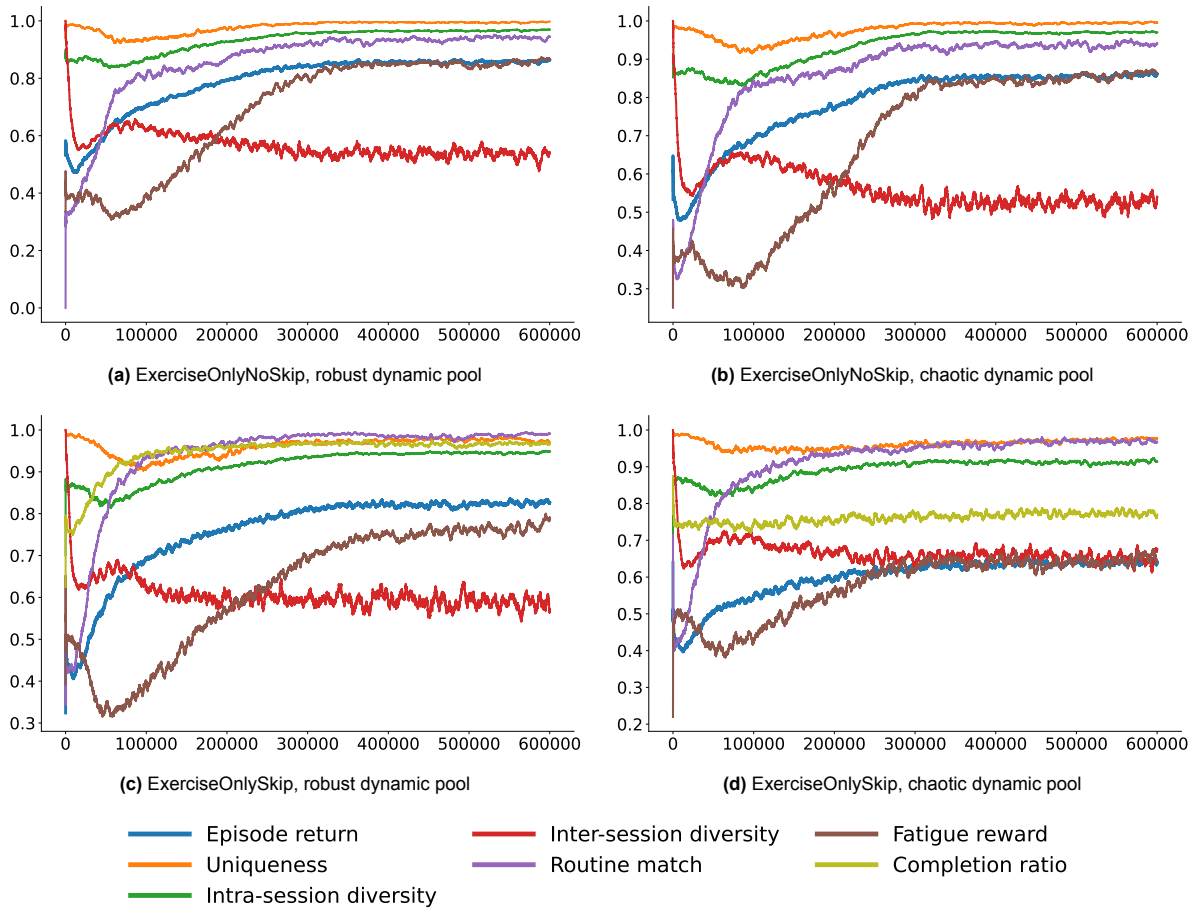


Figure B.9: Additional robustness training curves for the exercise-only environments. Shared color legends are shown below each row. These are the exercise-only counterparts of Figure 6.13.

C

Additional Illustrative Examples

C.1. Additional recommendation examples

This appendix provides two additional single-episode recommendation examples for the environments not shown in the main text. As in Section 4.8, each example is shown in three steps: first the sampled user and session context, then the recommendation itself, and finally the resulting reward scores. These examples are included only to make the environments more concrete.

C.1.1. `ExerciseOnlySkip` example

Tables C.1–C.3 show an illustrative recommendation from the `ExerciseOnlySkip` environment.

Feature	Value
User ID	286
Sex	female
Age	36
Height (cm)	166.16
Weight (kg)	50.53
Goal	endurance
Experience	advanced
Training frequency	1
Sampled session routine	pull day
Days since last session	6
True capacity C_{true}	1.080
Estimated capacity $\hat{C}$	1.123
Estimated skip bias b_{hat}	0.000
Base skip rate	0.0079

Table C.1: Sampled user and session context for the illustrative `ExerciseOnlySkip` recommendation.

C.1.2. `FullPrescriptionNoSkip` example

Tables C.4–C.6 show an illustrative recommendation from the `FullPrescriptionNoSkip` environment.

Step	Exercise	Target muscles	Secondary muscles	Equipment	Body part
1	dumbbell lying rear delt row	upper back	shoulders, biceps	dumbbell	back
2	lever back extension	spine	glutes, hamstrings	leverage machine	back
3	smith back shrug	traps	shoulders	smith machine	back
4	barbell wrist curl	forearms	biceps, brachialis	barbell	lower arms
5	barbell pullover	lats	chest, triceps	barbell	back
6	hyperextension	spine	glutes, hamstrings	body weight	back
7	dumbbell shrug	traps	shoulders	dumbbell	back
8	barbell one arm bent over row	upper back	biceps, forearms	barbell	back

Table C.2: Illustrative recommendation produced by PPO in the `ExerciseOnlySkip` environment.

Metric	Value
Total reward	0.968
R_{unique}	1.000
R_{intra}	0.958
R_{inter}	1.000
R_{routine}	1.000
R_{fatigue}	0.882
Completion ratio	1.000

Table C.3: Terminal reward and component scores for the illustrative `ExerciseOnlySkip` recommendation.

Feature	Value
User ID	307
Sex	male
Age	56
Height (cm)	187.18
Weight (kg)	98.88
Goal	fat loss
Experience	beginner
Training frequency	5
Sampled session routine	leg day
Strength multiplier	1.000

Table C.4: Sampled user and session context for the illustrative `FullPrescriptionNoSkip` recommendation.

Step	Exercise	Sets	Reps	Load (kg)	Target muscles	Secondary muscles	Equipment	Body part
1	decline crunch	3	20	0.5	abs	hip flexors	body weight	waist
2	lever seated hip abduction	3	15	20.0	abductors	glutes, hamstrings	leverage machine	upper legs
3	dumbbell standing calf raise	3	12	7.0	calves	ankles	dumbbell	lower legs
4	mountain climber	3	20	0.5	cardiovascular system	core, shoulders, triceps	body weight	cardio
5	lever seated hip abduction	3	12	25.5	abductors	glutes, hamstrings	leverage machine	upper legs
6	lever seated crunch	3	12	19.5	abs	obliques	leverage machine	waist
7	lever leg extension	3	10	28.0	quads	hamstrings	leverage machine	upper legs
8	lever standing calf raise	4	10	29.5	calves	soleus, ankle stabilizers	leverage machine	lower legs

Table C.5: Illustrative recommendation produced by PPO in the `FullPrescriptionNoSkip` environment.

Metric	Value
Total reward	0.911
R_{unique}	1.000
R_{intra}	0.975
R_{inter}	1.000
R_{routine}	1.000
R_{fatigue}	0.905
R_{rx}	0.924
R_{persist}	0.780

Table C.6: Terminal reward and component scores for the illustrative `FullPrescriptionNoSkip` recommendation.

D

Data Search for Gym Recommendation Data

This appendix summarizes the broader data search that was carried out before the final simulator-based setup was chosen. The search was useful during the thesis because the final design depends strongly on what kind of public data is actually available for gym workout recommendation.

For the recommendation problem studied in this thesis, the ideal data would combine several elements at once: a broad gym exercise catalog, session-level exercise order, prescription variables such as sets, repetitions, and load, user-specific context, and some form of interaction or adherence signal. In other words, the required data is both about what exercises exist and about what users actually did, under which circumstances, and how they responded over time.

D.1. (Lack of) User Interaction Data at the Gym

The clearest pattern from the data search is that publicly available gym-related data is fragmented. Many datasets contain useful information for one part of the problem, but very few contain the combination of exercise content, session structure, prescription variables, and user interaction that a sequential gym recommender would need.

A first category consists of content-only exercise datasets. These datasets provide structured information about gym exercises, such as movement patterns, target muscles, equipment, and short descriptions. They are useful for defining the exercise catalog and for attaching semantic features to candidate actions, but they do not record what users actually performed in real sessions. As a result, they are suitable for representing the exercise space, but not for learning session adaptation or personalized prescription behavior. Sources in this category include ExerciseDB V1 [18], Wrkout Exercises [79], the Free Exercise DB [32], Gym Exercises Datasets [38, 36, 37], the MuscleWiki site and a MuscleWiki clone [53, 52], StrengthLog [72], the Functional Fitness Exercise Database [33], and the Muscle & Strength Exercise Video Database [50].

A second category consists of workout plans, routine templates, and preference resources. These sources are useful because they show how gym sessions are typically structured. They can therefore inform the logic of routines, training goals, and combinations of exercises that commonly appear together. However, they still do not contain user interaction data. They describe what a session could look like, but not how a specific user followed, modified, or skipped such a session. Examples include Muscle & Strength Workout Routines [51] and Workout Preferences and Fitness Goals [77].

A third category is executed workout logs. These come closest to the recommendation problem in this thesis, because they contain traces of actually performed sessions. The 721 Weight Training Exercises dataset [1], for example, provides roughly 10,000 executed sets including date, exercise, set order, repetitions, load, and rest. This makes it valuable because it captures several variables that matter directly for gym prescription. At the same time, its usefulness is limited by the fact that it

is essentially a single-person log, focused on a narrower set of exercises, and without broader user-context information. It is therefore helpful as an example of what executed gym data can look like, but not sufficient as a training and evaluation source for a personalized recommender.

FitRec [31] also contains large sequential workout data, but the domain is different. It includes workout trajectories with heart rate and GPS information, yet is centered mainly on endurance activities such as running and cycling rather than resistance training in the gym. This makes it useful as evidence that sequential exercise data can exist at scale, but not as a direct match for the gym recommendation problem studied here. Other sources, such as forum-style logs [69], show that some workout history data exists in practice, but these records are often noisy, incomplete, unstructured, or difficult to use reliably in a recommendation pipeline. In addition, some relevant datasets were described in papers but were not publicly released [46, 75].

A fourth and much larger category consists of sensor, repetition, and form-quality datasets. These are currently the most common type of exercise-related data in the gym domain. Such datasets are useful for recognizing exercises, counting repetitions, estimating intensity, or giving feedback on execution quality. Examples include RecGym [56], Gym Workout IMU Dataset [39], Exercise Recognition from Wearable Sensors [17], and Sensor-Based Gym Physical Exercise Recognition [63]. Multimodal and video-based resources such as MM-Fit [48], RepCount [57], Fit3D [28], and Fitness-AQA [30] are also valuable for repetition counting, movement analysis, and form assessment.

These datasets are highly relevant for exercise recognition and coaching, but they solve a different problem from the one studied in this thesis. In most cases they capture how an exercise was performed, not what should be recommended next. They usually lack broader session context, user history, goal information, and interaction traces over multiple workout decisions. This makes them useful for verification and form feedback, but not for a recommender that constructs full gym sessions over time.

There are also generic health and tracker datasets, but these are usually too broad or too weakly tied to gym behavior to support the present task [29]. They may contain step counts, heart rate, or general activity levels, but not the exercise-by-exercise and set-by-set structure needed for gym recommendation.

Taken together, the data search points to a clear gap. Public data is available for exercise content, for routine examples, for sensor-based recognition, and to a limited extent for executed workout logs. What is largely missing is a dataset that combines all of the information needed for a context-aware sequential gym recommender: a sufficiently broad gym exercise catalog, session-level exercise order, prescription variables, user context, and realistic interaction or adherence feedback. This gap is one of the main reasons why the thesis uses synthetic users and a simulator instead of a standard logged-data recommendation setup.