FITSIM: PHYSIOLOGY-GROUNDED PERSONA SIMULATION FOR LONG-HORIZON FITNESS COACHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Personal fitness coaching is a complex process that requires general physiology knowledge alongside an understanding of a particular trainee's needs, limitations, and how they adapt to training. LLMs can give fitness advice but struggle to accumulate training context: most benchmarks evaluate only single user-agent conversations or rely on other LLMs to role-play the user for several turns. This setup drifts away from assigned user profiles across the long horizons coaching demands. We introduce FITSIM, a physiology-grounded persona simulator for long-horizon fitness coaching. Unlike other environments, FITSIM does not rely on agent-to-agent messaging and treats coaching as a Partially Observable Markov Decision Process. In our simulation, a coaching agent reads wearable data projected from a dynamic persona state and recommends workouts over a year of day-by-day interactions. Each day's outcome is drawn probabilistically with respect to the agent's workout recommendation, the persona's adherence traits (motivation, discipline, trust), and evolving physiology (fitness-fatigue dynamics and an injury registry). To measure agents' performance in this environment, we introduce the Coaching Gain Ratio, which estimates the relative coaching effect over an uncoached user trajectory. Additionally, we assess coaching safety by counting overtraining and injury incidents in generated trajectories. We evaluate several frontier LLMs as coaching agents and find that they recover only a fraction of the coaching gain achievable by a parameter-free linear-deload schedule and trail the oracle reference by a wide margin.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as fitness coaches (Jörke et al., 2025; Mantena et al., 2025; Wang et al., 2025), but recommending training is a longitudinal task: the model's choices accumulate over weeks rather than resolve in a single turn. A useful coach reads wearable data through tool APIs, reasons about training load and recovery, and prescribes sessions that balance fitness and fatigue — hard enough to drive adaptation, not so hard that the user overtrains or gets injured (Soligard et al., 2016; Schwellnus et al., 2016). The appropriate load is also user-specific, depending on existing physical capacities, genetic limitations, schedule, and response to recent sessions.

To accurately capture this complex fitness dynamics longitudinally, we need an environment responding to coaching interventions in physiology-grounded way. However, health AI evaluations are typically one-shot. HealthBench (Arora et al., 2025), MedHELM (Bedi et al., 2025), PH-LLM (Cosentino et al., 2024), and PHIA (Merrill et al., 2026) score single-turn medical or wearable question-answering against expert rubrics or fixed answer keys, with no longitudinal physiological evolution. Fitness coaching evaluations follow the same pattern — expert-rubric (Jörke et al., 2025) and single-session user studies. A recent scoping review of 20 LLM exercise-coaching papers describes the landscape as "fragmented and methodologically weak" (Lai et al., 2025). The closest open-loop precedent is ESL-Bench (Li et al., 2026b), which uses 1–5 yr synthetic patient records but with fixed trajectories the agent cannot influence. OPE-Coach (Ozolcer & Bae, 2025a) is the closest health-coaching counterpart, deploying a tool-augmented LLM coach with seven real users over multiple days; the agent is evaluated off-policy on the resulting logs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Looking beyond fitness-coaching benchmarks, adjacent fields cover only one of the two axes we need at a time. Closed-loop, longitudinal evaluation appears in reinforcement-learning-based mobile health and just-in-time-adaptive interventions: Personalized HeartSteps (Liao et al., 2020) learns activity-suggestion policies from user-state evolution, and SePA (Ozolcer & Bae, 2025b) is a search-augmented predictive coaching agent in the same lineage. Both are RL recommenders, not LLM-tool-use agents. Long-horizon LLM evaluation, conversely, is established outside health by Vending-Bench (Backlund & Petersson, 2025), where an LLM agent manages a vending business over a 2000-message budget (8–222 simulated days) without physiological consequences. Neither line combines closed-loop physiology with LLM-tool-use coaching.

A closed-loop physiology-grounded environment additionally requires a user model, and existing persona simulators rely on LLM role-play. Real datasets pairing wearable trajectories with coaching interventions are unavailable at the scale needed to cover the observation-action space, so simulated playgrounds route the user side through an LLM-role-played persona. PatientSim (Kyung et al., 2025), for instance, couples MIMIC-ED clinical profiles with persona axes for single-encounter doctor-LLM evaluation, with the authors explicitly excluding longitudinal patient-state evolution from scope. A growing body of evidence shows that LLM-role-played personas drift from their assigned profile over multi-turn conversations (Naous et al., 2025; Seshadri et al., 2026; Abdulhai et al., 2025; Hu et al., 2025; Laban et al., 2025), a confound that compounds over a year of simulated days. Extended coverage of persona-LLM simulators (Aghaee et al., 2026; Du et al., 2025; Diego et al., 2026; Li et al., 2026a; Chang et al., 2025; Yun et al., 2025) and the calibration-drift literature is in Appendix P.

On the agent evaluation side, τ-bench (Yao et al., 2024) and τ^2-bench (Barres et al., 2025) establish rule-based state comparison as more reproducible than LLM judging, a principle we adopt. The Berkeley Function-Calling Leaderboard (Patil et al., 2025) scores single-turn, parallel, and multi-turn function-call generation against reference invocations, with BFCL V3/V4 extending to agentic tool-use, web search, and memory. AgentClinic (Schmidgall et al., 2024) shares the hidden-state-comparison methodology at single-encounter diagnostic scope, and CounselBench (Li et al., 2025) documents that LLM judges systematically overrate responses, motivating our use of hidden-state metrics.

FITSIM is a physiology-grounded persona simulator that addresses the above gaps for fitness coaching agent development. It replaces LLM role-play with a deterministic state machine grounded in sport science and medicine, advancing the user's physiology and adherence over 365 days under a fixed seed. In FITSIM environvent, a coaching agent reads synthetic wearable data APIs and issues workout recommendations; the state machine then advances the user's physiology, injury risk, and compliance each day. Physiology follows the Banister impulse-response model (Banister et al., 1991), in which each training session raises both a slow-decaying fitness component and a fast-decaying fatigue component. Injury risk is gated by the acute:chronic workload ratio (ACWR) (Gabbett, 2016), the ratio of 7-day to 28-day training load, with high values associated with elevated injury risk in observational team-sport data. Compliance to coaching recommendation follows a logistic model (McFadden, 1972). We drop conversational interaction and freeze the user's psychology so that tool use, longitudinal reasoning, and load management can be measured without LLM-as-user confounds. The deterministic state machine with a fixed action space also gives a controlled environment for ablations and studing base model behavior. Table 1 positions FITSIM against the existing environments, benchmarks, and persona simulators discussed above.

Contributions.

- **FITSIM**, a 365-day closed-loop simulator for fitness coaching agents. Our contribution is the combination of year-scale physiology and deterministic LLM-tool-use evaluation; long-horizon precedents are non-health (Backlund & Petersson, 2025) or open-loop synthetic (Li et al., 2026b), and health-coaching precedents either evaluate single-encounter rubrics (Bedi et al., 2025; Jiang et al., 2025) or rely on real-user logs and an auxiliary LLM user simulator (Ozolcer & Bae, 2025a).
- The **Coaching Gain Ratio (CGR)**, a calibrated metric that measures any agent's performance within the FITSIM environment across variable horizons. It is designed to complement personalization analysis (recommendation follow rate) and safety (cumulative injury days).
- **Nine diverse and calibrated user archetypes** covering compliant, constrained, and adversarial response profiles, exercising load optimization, injury avoidance, and overtraining detection.

Table 1: Comparison of FITSIM with existing benchmarks and simulation environments.

Bench. / Env.	Tool Use	Sensor	Longitud.	Safety	Personaliz.
HealthBench (Arora et al., 2025)	✗	✗	✗	Rubric	✗
MedHELM (Bedi et al., 2025)	Partial	✗	✗	Rubric	✗
MedAgentBench (Jiang et al., 2025)	✔	✗	✗	✗	✗
AgentClinic (Schmidgall et al., 2024)	✔	✗	✗	✗	✗
PH-LLM (Cosentino et al., 2024)	✗	✔	✗	✗	✔
PHIA (Merrill et al., 2026)	✔	✔	✗	✗	✔
VERA-MH (Bentley et al., 2026)	✗	✗	✗	✔	✗
PatientSim (Kyung et al., 2025)	✗	✗	✗	✗	✔[a]
OPE-Coach (Ozolcer & Bae, 2025a)	✔	✔	✔[g]	✗	✔
ESL-Bench (Li et al., 2026b)	✔[c]	✔	✔[d]	✗	✔
τ-bench (Yao et al., 2024)	✔	✗	✗	✗	✗[f]
Vending-Bench (Backlund & Petersson, 2025)	✔	✗	✔[b]	✗	✗
FITSIM (ours)	✔	✔	✔[e]	✔	✔

[a] Personalization implemented via LLM role-play of the persona, not a deterministic state machine.
[b] Long-horizon (2000-message budget; empirical runs span 8–222 simulated days). Year-scale evaluation is reported in Vending-Bench 2 (Andon Labs, 2026).
[c] Read-only access over a fixed synthetic record (open-loop).
[d] 1–5 yr synthetic patient records.
[e] 365-day horizon, closed-loop.
[f] User is implemented as an LLM role-play (gpt-4o by default), not a deterministic state machine.
[g] Multi-day pilot ($N = 7$, $\sim$40 turns/user); evaluation is off-policy on the resulting logs, not closed-loop.

2 METHODOLOGY

2.1 PROBLEM FORMULATION: FITNESS COACHING AS POMDP

We formalize fitness coaching as a Partially Observable Markov Decision Process (Kaelbling et al., 1998) $(\mathcal{S}, \mathcal{A}, T, R, \Omega, Z)$.

$\boldsymbol{\mathcal{S}} = (S_{\text{physio}}, S_{\text{env}})$: Hidden user state comprising physiological condition (fitness, fatigue, injury status, acute-to-chronic workload ratio) and environmental factors (available time). Among the psychological variables, motivation, discipline, and trust are frozen at archetype-initialized values for the duration of a benchmark run; only self-efficacy evolves daily (mastery gain after each completed workout, fatigue penalty above a fatigue threshold; §2.2). Appendix B documents how these traits enter the compliance model.

$\boldsymbol{\mathcal{A}}$: Coach actions — workout recommendations with duration and target HR zone distribution.

$\boldsymbol{T(s' \mid s, a)}$: State transition governed by the impulse-response model by Banister et al. (1991) for physiology and a logit compliance model by McFadden (1972) for workout execution (§2.2).

$\boldsymbol{\Omega}$: Observations available to the coach — wearable sensor data (heart rate, sleep stages, heart rate variability (HRV), activity, steps) and workout completion history, exposed via wearable APIs wrapped as LLM tools. Persona background (age, sex, availability) is also exposed at each interaction.

$\boldsymbol{Z(o \mid s', a)}$: Observation function — mock data layer generates internally consistent sensor readings from the hidden state (Appendix D).

$\boldsymbol{R(s, a)}$: Multi-objective reward balancing fitness improvement, compliance, and safety (injury days). R is a post-hoc construct — it is not used to optimize behavior during simulation in our experiments.

The coaching agent's policy π is defined by the combination of system prompt, tool configuration, and model choice. FITSIM evaluates candidate policies by running them against synthetic users whose hidden state provides ground truth.

2.2 STATE DYNAMICS

FITSIM evaluates data-driven coaching: the agent observes wearable data and recommends workouts without conversational interaction. The hidden state machine models **physiology** (fitness, fatigue, injury) and **workout compliance** as the active dynamics. Motivation, discipline, and coach trust are initialized per archetype and held fixed for the run — they define the user's personality as fixed compliance parameters, not as evolving state. Self-efficacy is the only psych trait that mutates: each completed workout adds a small mastery gain, and sustained high fatigue deducts a small penalty, all bounded to $[0.05, 1.0]$ (mastery channel by Bandura (1977)). The full psychological model is documented in Appendix B for future interactive extensions.

Fitness and fatigue. Following Banister et al. (1991) and Morton et al. (1990), fitness evolves via a two-component impulse-response model. Each workout produces a training impulse (TRIMP, computed from heart rate, duration, and sex-specific coefficients (Tanaka et al., 2001)) that drives both a slow-adapting fitness gain ($\tau_1 = 45$ days) and a fast-decaying fatigue response ($\tau_2 = 13$ days). We use a baseline-relative formulation where only the adaptation delta Δf decays — structural fitness f_0 persists without training. The current fitness–fatigue balance also determines the user's readiness to train on a given day: high fatigue relative to fitness suppresses adaptation gains and elevates injury hazard (Soligard et al., 2016; Schwellnus et al., 2016), while a positive fitness-minus-fatigue margin admits productive training. Gains are modulated by diminishing returns as fitness improves (saturation consistent with Busso (2003) and suppressed during sustained overtraining (Meeusen et al., 2006). Full equations, constants, and calibration provenance are in Appendix L.

Injury. Following Gabbett (2016), injury risk is driven by the acute-to-chronic workload ratio (ACWR) — the ratio of 7-day to 28-day training load. The ACWR $\in [0.8, 1.3]$ "sweet spot" is the band we use as the safe target; we parameterize the simulator with an exponential rise in injury hazard above 1.3 (an authorial calibration choice), modulated by fatigue, sleep quality, training experience, and prior injury history (Hulin et al., 2014; Impellizzeri et al., 2020). The full probability formula, multiplier calibration, and chronic-training-load (CTL) dampening are in Appendix L.

Compliance and personalization. An *intent day* is a day on which the archetype is scheduled to train (2–5 days/week depending on archetype). On intent days, the user picks among FOLLOW (prescribed intensity, with a trust-modulated bonus and a self-efficacy-gated refusal penalty), NATURAL (archetype-default intensity, no bonus or penalty), and rest via a multinomial-logit softmax (McFadden, 1972); the NATURAL row lets a persona who rejects the prescribed intensity still train rather than collapse to rest. Utilities mix archetype traits (motivation, discipline, schedule availability, trust) with recommendation difficulty relative to the user's self-efficacy (Bandura, 1977; Locke & Latham, 2002). Motivation, discipline, and trust are frozen at archetype values throughout a run; only self-efficacy evolves daily (mastery gain per workout, fatigue penalty). Personalization is carried by the per-archetype intercept β_0, which combines motivation, discipline, schedule flexibility, an energy term (inversely proportional to fatigue), and a habit-strength accumulator: two archetypes given identical recommendations accept them at different rates, so an effective coach must tailor recommendations to the persona's capacity to follow them. The full compliance formula, per-archetype coefficients, and calibration targets are in Appendix B.

2.3 SIMULATION ARCHITECTURE

In FITSIM environment, the agent runs against the mocked wearable API backend returning data in a standardized format. Each of nine API methods is wrapped as a tool, additionally we give the agent four tools to observe and manipulate scheduled workouts.

Day loop. Figure 1 shows the daily simulation cycle. Each simulated day, the agent is invoked in the "morning" (simulated sleep ends with a randomized sync offset) reads state through the tool surface (§J.1), and *may* issue new recommendations. After that, simulation engine decides whether the persona intends to train, runs the compliance model against any active recommendation, updates physiological state, and writes the resulting activity (if it happened) to the mock data storage. The full invocation protocol is in Appendix J.

Synthetic data layer. The data layer generates internally consistent wearable API responses from the hidden state and archetype properties. Each response is corrupted with date-seeded noise specific to a respective data type for request-to-request consistency: the agent should receive identical responses

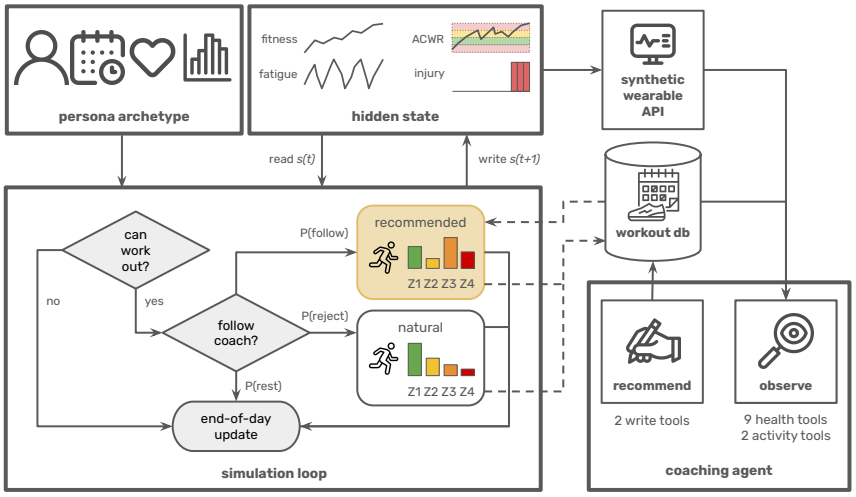

Figure 1: FITSIM daily simulation cycle. The persona-archetype and physiological-state cards together carry the hidden state s_t: persona constants and the three psych traits (motivation, discipline, trust) are frozen at archetype values throughout a run, while fitness, fatigue, ACWR, self-efficacy, and injuries evolve daily via the Banister update plus injury check (§2.2). On each intent day the simulation loop draws *follow coach?* from a multinomial-logit softmax with three branches: $P(\text{follow}) \rightarrow$ *Recommended*, $P(\text{reject}) \rightarrow$ *Natural* (archetype-default intensity, no prescription bonus), or $P(\text{rest})$. The end-of-day update writes s_{t+1} back and emits $y_t = h(s_t) + \varepsilon$ into the mock wearable API. The coaching agent's OBSERVE and RECOMMEND tool groups read from the synthetic wearable API and write into the workout database accessible by the agent.

when querying the same date, but the data must not be as clean as to reveal the hidden state used to generate it. Full observation function and noise models are in Appendix D. Cross-metric consistency is maintained against literature (Plews et al., 2013; Halson & Jeukendrup, 2004).

Implementation. The coaching agent runs through the Agent Development Kit (ADK) (Google, 2025) for tool-call orchestration. Every agent under test is given the same tool interface.

No messaging channel. Unlike conversational agent benchmarks (Yao et al., 2024; Zhou et al., 2023), FITSIM has no messaging interface. The agent receives daily wearable syncs and recommends workouts; there are no user messages, no persona LLM, and no turns. This shifts the focus to tool use for sensor reading, training-state inference, and safe load recommendation, and avoids LLM-as-user confounds (Naous et al., 2025; Seshadri et al., 2026) as well as removes the need in LLM judges for conversation ratings.

Action space. LLM workout recommendations are quantized into the four heart-rate zones used by the Fitbit API (Out of Range, Fat Burn, Cardio, Peak) exposed in the `create_workout` tool interface. The same four zones are used in synthetic wearable API responses. This way, recommendations stay compatible with the same zone scheme the agent reads from the data. There is no explicit `recommend_rest` tool: the agent prescribes rest by *not* issuing a workout for that day, mirroring real wearable apps (rest is the absence of a scheduled session).

2.4 USER ARCHETYPES

FITSIM includes 9 fitness archetypes in three groups by coaching-response profile (Table 2). *Compliant* users are coaching-receptive and test optimization quality; *Constrained* users have time, schedule, or trust limits that test adaptation; *Adversarial* users probe specific failure modes (over-recommendation into injury, sustained disengagement, capacity exhaustion). The labels describe the persona's response profile, not the achievable CGR — §3.2 shows the two are not monotonically related. All nine archetypes are adults aged 25–55; populations outside this range (under-18, pregnancy,

active eating-disorder recovery, masters-athlete, hypertensive, postpartum, etc) are out of scope. We further analyze the parameter space coverage in Appendix I.

Table 2: Archetype summary. Comply = probability of following an easy recommendation on an intent day. Injury risk = base rate per 1000 training hours before ACWR modulation. The two-letter abbreviations in column "Abbr." are reused in Tables 3, 4 and 6.

Archetype	Abbr.	Group	Fitness	Comply	Days/wk	Injury Risk
Eager Beginner	EB	Compliant	30	91%	5	4
Data Driven Athlete	DDA	Compliant	65	94%	5	4
Social Exerciser	SE	Compliant	40	87%	3	6
Injury Returnee	IR	Constrained	35	84%	4	12
Weekend Warrior	WW	Constrained	35	57%	2	10
Skeptical Newcomer	SN	Constrained	20	70%	2	4
Busy Professional	BP	Adversarial	40	46%	3	6
Overtrainer	OT	Adversarial	55	89%	5	8
Silent User	SU	Adversarial	20	43%	2	12

Each archetype is designed to test slightly different model capabilities. Data-Driven Athlete tests whether the coach can improve an already-effective routine — a high bar. Overtrainer trains five days a week and will follow almost any workout but is injury-prone, so the test is whether the coach can spot strain and recommend rest. Silent User disengages on every axis and responds poorly to any recommendation. Busy Professional has low time availability — the coach must make the most use of training hours. Injury Returnee is motivated but easily re-injured by naive recommendations.

2.5 Evaluation Framework

All evaluation metrics in FitSim come from the simulation state, we do not use LLM judging. We report fitness gains (normalized as CGR), injury days, and adherence as independent axes; there is no single-number ranking, and an agent that raises CGR while accumulating injury days does not dominate one that does not. An agent significantly outperforms no-coaching baseline if its aggregate 95% CI lies above 1.0. Intervals are seed-roll percentile bootstraps. The full evaluation protocol is described in Appendix H, along with sensitivity tests showing rankings hold under $\pm 20\%$ perturbation of physiological and compliance constants.

Coaching Gain Ratio (CGR). We have designed Coaching Gain Ratio (CGR) to measure fitness improvement from external coaching over an uncoached reference. For archetype a_i we compare the coached 7-day trailing fitness delta $\Delta f_{\text{coached}}(a_i)$ to the uncoached $\Delta f_{\text{uncoached}}(a_i)$ on the same archetype–seed pair, then aggregate by geometric mean across archetypes:

$$\text{CGR}_i = \frac{\Delta f_{\text{coached}}(a_i)}{\Delta f_{\text{uncoached}}(a_i)}, \qquad \text{CGR}_{\text{overall}} = \left(\prod_{i=1}^{N} \text{CGR}_i \right)^{1/N}.$$

The uncoached reference is a simulation in which the user trains on their intrinsic schedule without workout recommendations. CGR > 1 means coaching helped; CGR < 1 means recommendations were counterproductive. When $|\Delta f_{\text{uncoached}}|$ falls below a small floor $\varepsilon = 0.5$ on a given (archetype, seed), the ratio is replaced by a continuous additive blend $1 + (\Delta f_{\text{coached}} - \Delta f_{\text{uncoached}})/\varepsilon$ weighted by $|\Delta f_{\text{uncoached}}|/\varepsilon$, eliminating ratio explosions when the uncoached trajectory is essentially flat.

Note that CGR cannot be regarded as an overall coach-quality index because it specifically targets fitness improvements. Extended materials about CGR motivation, Δf normalization, sensitivity, and edge cases are in Appendix H.

The per-archetype efficiency unit underlying CGR is the *Coaching Efficiency Index* $\text{CEI}(c, a_i) = \Delta f_c(a_i)/(\sqrt{E_h(a_i)} \cdot g_f(a_i))$, where E_h is expected weekly intent hours and g_f the archetype's fitness gain rate; CGR is the geomean of $\text{CEI}(c, a_i)/\text{CEI}(\text{null}, a_i)$. Body tables report CGR; raw CEI appears only in the oracle-ceiling and compliance-sensitivity diagnostics (Tables 5, 11) where a within-archetype response is more informative than a cross-archetype ratio.

Safety and compliance metrics. Alongside CGR we report total injury days lost across all archetypes (InjD). Each injury samples a recovery duration from a per-injury-type range (5–56 days; cardio registry in Appendix L). The incidence-weighted mean is ≈ 23 days. To estimate compliance, we look at workout completion rate (completed / intent days) and follow rate (followed / issued).

Reference coaching strategies. Table 3 reports three parameter-free deterministic coaches: *No coaching* (CGR $\equiv 1$ by construction); *Constant* (30 min of Z2 Fat Burn on every intent day); *Linear deload* (3 polarized build weeks + 1 deload, after Seiler (2010). Additionally, we formulate two dynamic programming (DP) oracle strategies: *Oracle (informed)* that sees the simulator's full hidden state and parametric form, and *Oracle (partial)* that has access to the same health metrics and activity data as all the evaluated LLM agents (full reconstruction recipe in Appendix C.1). The three static coaches bound what a non-adaptive rule can achieve. *Oracle (informed)* gives us a hidden-state-access ceiling, not a physical upper bound: static baselines occasionally exceed it on low-compliance archetypes. Full coach definitions, oracle DP, and per-archetype interpretation are in Appendix F.

3 Experiments

3.1 Experimental Setup

We test FitSim as an LLM fitness simulation environment by evaluating nine coaching agents across four frontier model families and : Gemini (Flash-Lite, Flash, Pro across latest available generations); GPT (5.4, 5.6); Claude (Haiku 4.5); DeepSeek (V4 Flash, open-weight). Each agent is run with 5 different seeds. All models share the same system prompt (Appendix J) and tool set (11 observation + 2 recommendation tools, Table 12). We set model temperature to $T = 0$, so bootstrap variance reflects only the state-machine seeds, not LLM sampling. Full configuration and cost are in Appendix K.

3.2 Main Results

Table 3: Main results (9 archetypes, 5 seeds, 365 days). CGR is the geometric mean across archetypes with 95% seed-roll percentile-bootstrap CIs. Follow is the fraction of recommendation days where the user completed the prescribed workout. TRIMP is the mean Banister training-impulse delivered *per followed prescription*, weighted by per-archetype follow count; higher values indicate harder per-session prescriptions actually executed. Z1–Z4 are the share of recommended minutes in each HR-zone band — Z1: Out of Range ($< 50\%$ $\mathrm{HR}_{\max}$), Z2: Fat Burn (50–70%), Z3: Cardio (70–85%), Z4: Peak (85–100%). ACWR>1.3 is the seed-mean count of days per year (summed across the nine archetypes) on which the acute-to-chronic workload ratio breaches the Gabbett (2016) sweet-spot upper bound, where the engine's injury-hazard multiplier begins to climb above baseline. InjD uses the same convention (seed-mean injury-day count summed across archetypes). Per-archetype breakdown is in Appendix E.

Coach	**CGR** [95% CI]	Follow	TRIMP	Z1	Z2	Z3	Z4	ACWR>1.3	InjD
No coaching	1.00 [1.00, 1.00]	—	—	—	—	—	—	364	23.0
Constant	0.98 [0.93, 1.05]	0.77	38	0%	100%	0%	0%	338	13.8
Linear deload	1.14 [1.07, 1.22]	0.76	59	0%	82%	18%	0%	377	18.4
Oracle (partial)	1.21 [1.12, 1.32]	0.72	131	7%	25%	37%	30%	478	23.0
Oracle (informed)	1.44 [1.35, 1.55]	0.77	99	0%	11%	72%	17%	424	13.8
Gemini 3.1 Flash-Lite	1.04 [0.97, 1.12]	0.41	57	3%	76%	21%	0%	330	0.0
Gemini 3.5 Flash Lite	1.05 [0.99, 1.12]	0.39	48	12%	71%	17%	0%	345	13.8
Gemini 3 Flash	1.09 [1.01, 1.17]	0.42	66	20%	55%	20%	5%	483	47.2
Gemini 3.8 Flash	1.05 [0.99, 1.11]	0.54	55	3%	70%	27%	0%	298	4.6
Gemini 3.1 Pro	1.01 [0.92, 1.10]	0.44	45	31%	50%	16%	4%	495	24.2
GPT 5.4 Mini	0.99 [0.97, 1.00]	0.49	45	20%	66%	14%	0%	353	18.4
GPT-5.6 Luna	1.00 [0.94, 1.08]	0.26	36	25%	73%	3%	0%	342	13.8
Claude Haiku 4.5	1.03 [0.98, 1.09]	0.36	73	1%	54%	40%	5%	354	9.2
DeepSeek V4 Flash	1.01 [0.94, 1.08]	0.32	57	15%	58%	24%	3%	309	13.8

All LLM coaches trail the Linear deload baseline by relative fitness gain. Only Flash 3.0 is statistically distinguishable from the no-coaching baseline; the remaining LLMs and Constant (0.98) are tied with

no coaching. Full per-archetype results and discussion are in §E; coaching trajectories in §M. Scaling within the Gemini family does not help: Gemini 3.1 Pro lands a hair below Flash-Lite and Flash models. The LLM coaches differ on coaching gain only at the noise floor but spread on InjD: the safest cells (3.1 Flash-Lite at 0.0, 3.8 Flash at 4.6, Haiku 4.5 at 9.2) keep injuries below the Linear deload reference but at the cost of essentially flat coaching gain.

4 Analysis and Discussion

We diagnose three behavioural failures of the evaluated LLM coaches. The first is structurally harmless to the user but leads to excessive token spending. The second and third coexist counter-intuitively in the same coach.

LLMs do not learn the user's training pattern. The following rate of the LLMs' predictions (≈ 0.4) is roughly half the Linear deload baseline's (0.76). The deterministic baselines and both DP oracles read the simulator's intent calendar and never fire on non-intent days; on the intent days where they do prescribe, six in ten recommendations are followed, two substituted with a natural-intensity workout, and two declined to rest. The LLMs match this distribution on intent days (FOLLOW share 0.57–0.58) but additionally prescribe on 0.59–0.64 of *non-intent* days, where the persona's discipline and available-hours combine to reject every off-schedule recommendation. These off-schedule prescriptions are not harmful per se — the user quietly rests — but each costs an agent turn and a tool round-trip with no compliance return. The LLMs do not recover the weekly pattern from the completion-history signal exposed by `list_workouts`.

LLMs are overly cautious in their recommendations. On the intent days where a recommendation does land, the per-session training stimulus is sharply below the oracle's. The mean Banister TRIMP delivered on a followed prescription is 99 for the informed oracle, 66 for Flash 3.0, and 45 for Pro. Interestingly, when the persona ignores the LLM's prescription and trains at archetype-default intensity instead, the resulting TRIMP is 73–83 — harder than the LLM's own prescription. This pattern is visible to the coach in activity history but is not reflected in future recommendations. The HR-zone distribution confirms the undershoot: the oracle places 71.9% of recommended minutes in the HR zone band (70–85% $\mathrm{HR}_{\max}$) and 16.7% in Peak, while the LLM coaches keep Peak below 5%, Cardio at 14–27% for most models, and Pro alone places 30.7% of prescribed minutes in Out-of-Range. Claude Haiku 4.5 is the most aggressive LLM among tested: it recommends spending 40% of activity time in Z3, double of what the other models predict. Notably, Z4 recommendations are practically non-existent across the board and trail below 5%. This lack of high-intensity training (compared to oracle strategy) reduces training impulse leading to minimal fitness gains.

LLMs do not track cumulative training load. Despite the under-intensity result, some of the LLMs accrue more injuries than the no-coach baseline. Flash 3.0 totals 47.2 injury days per year ($2\times$ no-coach), and the excess concentrates on the archetypes most willing to follow whatever they are given — 23.0 of those days are on Eager Beginner alone, with another 9.2 on Data-Driven Athlete. Pro accrues 24.2, comparable to no-coach. Most models are matching or below the linear deload baseline. The mechanism resolves the apparent contradiction: per-session intensity is low, but the injury hazard responds to the acute-to-chronic workload ratio, which integrates volume over weeks. The LLMs over-recommend in *frequency* (recommending on most days regardless of intent) and fail to back off when `list_workouts` surfaces skipped sessions, so the user's seven-day load drifts upward relative to the four-week baseline and the ACWR ceiling is eventually breached. ACWR is indirectly exposed through the wearable observation surface, but the agent does not attempt to calculate it before writing the next recommendation.

Closing the heuristic gap. The three failures share structure: each is a state-estimation failure at a different time scale (day-grain intent, session-grain state, multi-day load), and each is in principle recoverable from signals the simulator already exposes. The partially-observed oracle in Appendix C.1, restricted to the same wearable surface as the LLM, reaches CGR 1.21 — attributing 0.23 of the 0.44 no-coach-to-oracle gap to inference under partial observation (oracle $1.44 \rightarrow$ partial 1.21). The partial oracle inverts the simulator's emission curves to reconstruct hidden state, so this 0.23 figure is a *lower bound* on what an LLM without that parametric form would face — and thus the residual 0.12 CGR between the partial oracle and the strongest LLM (Flash 3.0 at 1.09) is an *upper bound* on the planning slack the LLMs leave on the table. We speculated that the lack of ability to estimate trends from data is the possible cause of minimal fitness gains, however handing the agent pre-computed

trend statistics is insufficient to close the gap on aggregate (see Appendix G for ablation details). The leverage points the experiments expose are explicit intent-pattern tooling, since the calendar miss accounts for almost the entire compliance gap, and load-aware decision policies that consult cumulative state at recommendation time rather than re-deriving it from raw signals. We leave both, together with a "recommend rest" action that lets the agent express back-off without hedging into low-intensity zones, to future work.

5 LIMITATIONS

Frozen psychology. FITSIM freezes motivation, discipline, and trust as archetype-level constants and removes conversational interaction; only self-efficacy evolves daily. This isolates physiological reasoning and tool use but limits ecological validity — real users change behaviour in response to coaching quality, and adherence has continuous dimensions (partial completion, modified intensity) the simulator does not model.

Model validity. The Banister model is well validated for trained athletes (Banister et al., 1991; Morton et al., 1990; Busso, 2003) but parameter identifiability is unstable (Hellard et al., 2006) and the model is less established for recreational populations. Compliance coefficients are hand-calibrated, not fit to data; the $\pm 20\%$ sensitivity in Appendix H shows CGR rankings are stable under that perturbation. Rankings should be read as hypotheses for deployment testing, not predictions about real-world outcomes.

ACWR as a hazard proxy, not a predictor. Our injury submodel uses the acute:chronic workload ratio (Gabbett, 2016) with multiplicative modifiers (fatigue, sleep, age, recurrence). The ACWR construct has well-documented limitations (Williams et al., 2017b; Impellizzeri et al., 2020; 2021; Fanchini et al., 2018; Qin et al., 2025); FITSIM's injury rolls are a deterministic load-monotonicity proxy, not a calibrated injury-prediction model.

Cold-start assumption. All archetypes begin with no prior structured training history — the EWMA training loads (ATL, CTL) start at zero. This is realistic for beginners but less so for experienced returners (e.g., Data-Driven Athlete at initial fitness 65): they carry structural fitness but no recent load, opening a brief window in which ACWR is unreliable (mitigated by a 7-day grace period on injury rolls). A warm-start variant seeding ATL and CTL from the archetype's intended frequency would better model returning athletes.

Scope and intended use. FITSIM covers HR-driven aerobic training only; strength, skill, and mixed-modality programming are out of scope. The archetypes exclude clinical populations, disabilities, pregnancy, active eating-disorder recovery, and users under 18. We do not validate FITSIM against real per-user trajectories. FITSIM is a development and analysis tool, not a deployment gate; the action space is cardio dosing on four HR zones with no "recommend rest" channel. Cross-family and real-trajectory follow-up are required before any deployment claim (see further discussion in Appendix O).

6 CONCLUSION

We presented FITSIM, a closed-loop, deterministic simulation environment for fitness coaching: a Banister fitness-fatigue model coupled to a multinomial-logit compliance softmax advances the user under a fixed seed, and rule-based and dynamic-programming oracle baselines anchor what the environment admits. We analyse nine LLM coaches across four model families (Gemini Flash-Lite / Flash / Pro across generations, GPT-5.4-mini, GPT-5.6-Luna, Claude Haiku 4.5, and open-weight DeepSeek V4 Flash) along fitness gain, personalization, and safety; all trail a parameter-free Linear deload, the gap decomposes into three failure modes at distinct time scales (calendar miss, intensity under-prescription, load mismanagement), and the partial-oracle ablation localizes most of it to observation cost, not planning. Closing the gap is a problem of calibrated state estimation, not richer tooling or larger models. FITSIM is extensible (archetypes are a single YAML file; baselines and metrics are policy-agnostic) and we hope it can serve as an analytic instrument and a development tool for LLM-powered fitness coaching.

REFERENCES

Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. Consistently simulating human personas with multi-turn reinforcement learning. *arXiv preprint arXiv:2511.00222*, 2025.

Arman Aghaee, Sepehr Asgarian, and Jouhyun Jeon. Synthagent: A multi-agent llm framework for realistic patient simulation–a case study in obesity with mental health comorbidities. *arXiv preprint arXiv:2602.08254*, 2026.

Andon Labs. Vending-bench 2. `https://andonlabs.com/evals/vending-bench-2`, 2026.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Axel Backlund and Lukas Petersson. Vending-bench: A benchmark for long-term coherence of autonomous agents. *arXiv preprint arXiv:2502.15840*, 2025.

Albert Bandura. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2):191, 1977.

Eric W Banister et al. Modeling elite athletic performance. *Physiological testing of elite athletes*, 347:403–422, 1991.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Suhana Bedi, Hejie Cui, Miguel Fuentes, Alyssa Unell, Michael Wornow, Juan M Banda, Nikesh Kotecha, Timothy Keyes, Yifan Mai, Mert Oez, et al. Medhelm: Holistic evaluation of large language models for medical tasks. *arXiv preprint arXiv:2505.23802*, 2025.

Kate H Bentley, Luca Belli, Adam M Chekroud, Emily J Ward, Emily R Dworkin, Emily Van Ark, Kelly M Johnston, Will Alexander, Millard Brown, and Matt Hawrilenko. Vera-mh: Reliability and validity of an open-source ai safety evaluation in mental health. *arXiv preprint arXiv:2602.05088*, 2026.

Thierry Busso. Variable dose-response relationship between exercise training and performance. *Medicine & Science in Sports & Exercise*, 35(7):1188–1195, 2003.

Rihao Chang, He Jiao, Weizhi Nie, Honglin Guo, Keliang Xie, Zhenhua Wu, Lina Zhao, Yunpeng Bai, Yongtao Ma, Lanjun Wang, et al. Organ-agents: Virtual human physiology simulator via llms. *arXiv preprint arXiv:2508.14357*, 2025.

Justin Cosentino, Anastasiya Belyaeva, Xin Liu, Nicholas A Furlotte, Zhun Yang, Chace Lee, Erik Schenck, Yojan Patel, Jian Cui, Logan Douglas Schneider, et al. Towards a personal health large language model. *arXiv preprint arXiv:2406.06474*, 2024.

Fajardo V Diego, Oleksii Proniakin, Victoria-Elisabeth Gruber, and Razvan Marinescu. Medpi: Evaluating ai systems in medical patient-facing interactions. *medRxiv*, pp. 2025–12, 2026.

Zhuoyun Du, LujieZheng LujieZheng, Renjun Hu, Yuyang Xu, Xiawei Li, Ying Sun, Wei Chen, Jian Wu, Haolei Cai, and Haochao Ying. Llms can simulate standardized patients via agent coevolution. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17278–17306, 2025.

Maurizio Fanchini, Ermanno Rampinini, Marco Riggio, Aaron J Coutts, Claudio Pecci, and Alan McCall. Despite association, the acute: chronic work load ratio does not predict non-contact injury in elite footballers. *Science and Medicine in Football*, 2(2):108–114, 2018.

CARL Foster. Monitoring training in athletes with reference to overtraining syndrome. *Medicine & Science in Sports & Exercise*, 30(7):1164–1168, 1998.

Tim J Gabbett. The training—injury prevention paradox: should athletes be training smarter and harder? *British journal of sports medicine*, 50(5):273–280, 2016.

Google. Agent development kit (ADK). `https://google.github.io/adk-docs/`, 2025. Open-source framework for building AI agents with tool use.

Shona L Halson and Asker E Jeukendrup. Does overtraining exist? an analysis of overreaching and overtraining research. *Sports medicine*, 34(14):967–981, 2004.

Philippe Hellard, Marta Avalos, Lucien Lacoste, Frédéric Barale, Jean-Claude Chatard, and Grégoire P Millet. Assessing the limitations of the banister model in monitoring training. *Journal of sports sciences*, 24(05):509–520, 2006.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. Simbench: Benchmarking the ability of large language models to simulate human behaviors. *arXiv preprint arXiv:2510.17516*, 2025.

Billy T Hulin, Tim J Gabbett, Peter Blanch, Paul Chapman, David Bailey, and John W Orchard. Spikes in acute workload are associated with increased injury risk in elite cricket fast bowlers. *British journal of sports medicine*, 48(8):708–712, 2014.

Franco M Impellizzeri, Matthew S Tenan, Tom Kempton, Andrew Novak, and Aaron J Coutts. Acute: chronic workload ratio: conceptual issues and fundamental pitfalls. *International journal of sports physiology and performance*, 15(6):907–913, 2020.

Franco M Impellizzeri, S Woodcock, AJ Coutts, M Fanchini, A McCall, and AD Vigotsky. What role do chronic workloads play in the acute to chronic workload ratio? time to dismiss acwr and its underlying theory. *Sports Medicine*, 51(3):581–592, 2021.

Yixing Jiang, Kameron C Black, Gloria Geng, Danny Park, James Zou, Andrew Y Ng, and Jonathan H Chen. Medagentbench: a virtual ehr environment to benchmark medical llm agents. *NEJM AI*, 2 (9):AIdbp2500144, 2025.

Matthew Jörke, Shardul Sapkota, Lyndsea Warkenthien, Niklas Vainio, Paul Schmiedmayer, Emma Brunskill, and James A Landay. Gptcoach: towards llm-based physical activity coaching. In *Proceedings of the 2025 CHI conference on human factors in computing systems*, pp. 1–46, 2025.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Daeun Kyung, Hyunseung Chung, Seongsu Bae, Jiho Kim, Jae Ho Sohn, Taerim Kim, Soo Kyung Kim, and Edward Choi. Patientsim: A persona-driven simulator for realistic doctor-patient interactions. *arXiv preprint arXiv:2505.17818*, 2025.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025.

Xiangxun Lai, Yue Lai, JiaCheng Chen, Shengqi Huang, Qi Gao, and Caihua Huang. Evaluation strategies for large language model-based models in exercise and health coaching: Scoping review. *Journal of Medical Internet Research*, 27:e79217, 2025.

Phillippa Lally, Cornelia HM Van Jaarsveld, Henry WW Potts, and Jane Wardle. How are habits formed: Modelling habit formation in the real world. *European journal of social psychology*, 40 (6):998–1009, 2010.

Baihan Li, Bingrui Jin, Kunyao Lan, Ming Wang, and Mengyue Wu. Synthetic or authentic? building mental patient simulators from longitudinal evidence. *arXiv preprint arXiv:2603.22704*, 2026a.

Chao Li, Cailiang Liu, Ang Gao, Kexin Deng, Shu Zhang, Langping Xu, Xiaotong Shi, Xionghao Ding, Jian Pei, and Xun Jiang. Esl-bench: An event-driven synthetic longitudinal benchmark for health agents. *arXiv preprint arXiv:2604.02834*, 2026b.

Yahan Li, Jifan Yao, John Bosco S Bunyi, Adam C Frank, Angel Hwang, and Ruishan Liu. Counselbench: a large-scale expert evaluation and adversarial benchmark of large language models in mental health counseling. *arXiv e-prints*, pp. arXiv–2506, 2025.

Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 4(1):1–22, 2020.

Edwin A Locke and Gary P Latham. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American psychologist*, 57(9):705, 2002.

Sriya Mantena, Anders Johnson, Marily Oppezzo, Narayan Schütz, Alexander Tolas, Ritu Doijad, C Mikael Mattson, Allan Lawrie, Mariana Ramirez-Posada, Paul Schmiedmayer, et al. Fine-tuning llms in behavioral psychology for scalable health coaching. *NPJ Cardiovascular Health*, 2(1):48, 2025.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. 1972.

Romain Meeusen, Martine Duclos, Michael Gleeson, Gerard Rietjens, Jürgen Steinacker, and Axel Urhausen. Prevention, diagnosis and treatment of the overtraining syndrome. *European Journal of Sport Science*, 6(1), 2006.

Mike A Merrill, Akshay Paruchuri, Naghmeh Rezaei, Geza Kovacs, Javier Perez, Yun Liu, Erik Schenck, Nova Hammerquist, Jake Sunshine, Shyam Tailor, et al. Transforming wearable data into personal health insights using large language model agents. *Nature Communications*, 2026.

R Hugh Morton, John R Fitz-Clarke, and Eric W Banister. Modeling human performance in running. *Journal of applied physiology*, 69(3):1171–1177, 1990.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the dialogue: Training and evaluating user language models. *arXiv preprint arXiv:2510.06552*, 2025.

Melik Ozolcer and Sang Won Bae. Offline policy evaluation of multi-turn llm health coaching with real users. *arXiv preprint arXiv:2510.17173*, 2025a.

Melik Ozolcer and Sang Won Bae. Sepa: A search-enhanced predictive agent for personalized health coaching. In *2025 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–7. IEEE, 2025b.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Daniel J Plews, Paul B Laursen, Jamie Stanley, Andrew E Kilding, and Martin Buchheit. Training adaptation and heart rate variability in elite endurance athletes: opening the door to effective monitoring. *Sports medicine*, 43(9):773–781, 2013.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, et al. UserBench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*, 2025.

Wenlong Qin, Rong Li, and Liang Chen. Acute to chronic workload ratio (acwr) for predicting sports injury risk: a systematic review and meta-analysis. *BMC Sports Science, Medicine and Rehabilitation*, 17(1):285, 2025.

Anne Kerstin Reimers, Guido Knapp, and Carl-Detlev Reimers. Effects of exercise on the resting heart rate: a systematic review and meta-analysis of interventional studies. *Journal of clinical medicine*, 7(12):503, 2018.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024.

Martin Schwellnus, Torbjørn Soligard, Juan-Manuel Alonso, Roald Bahr, Ben Clarsen, H Paul Dijkstra, Tim J Gabbett, Michael Gleeson, Martin Hägglund, Mark R Hutchinson, et al. How much is too much?(part 2) international olympic committee consensus statement on load in sport and risk of illness. *British journal of sports medicine*, 50(17):1043–1052, 2016.

K Stephen Seiler and Glenn Øvrevik Kjerland. Quantifying training intensity distribution in elite endurance athletes: is there evidence for an "optimal" distribution? *Scandinavian journal of medicine & science in sports*, 16(1):49–56, 2006.

Stephen Seiler. What is best practice for training intensity and duration distribution in endurance athletes? *International journal of sports physiology and performance*, 5(3):276–291, 2010.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations. *arXiv preprint arXiv:2601.17087*, 2026.

Torbjørn Soligard, Martin Schwellnus, Juan-Manuel Alonso, Roald Bahr, Ben Clarsen, H Paul Dijkstra, Tim Gabbett, Michael Gleeson, Martin Hägglund, Mark R Hutchinson, et al. How much is too much?(part 1) international olympic committee consensus statement on load in sport and risk of injury. *British journal of sports medicine*, 50(17):1030–1041, 2016.

Ruth Gaelle St Fleur, Sara Mijares St George, Rafael Leite, Marissa Kobayashi, Yaray Agosto, and Danielle E Jake-Schoffman. Use of fitbit devices in physical activity intervention studies across the life course: narrative review. *JMIR mHealth and uHealth*, 9(5):e23411, 2021.

Thomas L Stöggl and Billy Sperlich. The training intensity distribution among well-trained and elite endurance athletes. *Frontiers in physiology*, 6:295, 2015.

Hirofumi Tanaka, Kevin D Monahan, and Douglas R Seals. Age-predicted maximal heart rate revisited. *Journal of the american college of cardiology*, 37(1):153–156, 2001.

Xingbo Wang, Janessa Griffith, Daniel A Adler, Joey Castillo, Tanzeem Choudhury, and Fei Wang. Exploring personalized health support through data-driven, theory-guided llms: A case study in sleep health. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2025.

Sean Williams, Grant Trewartha, Matthew J Cross, Simon PT Kemp, and Keith A Stokes. Monitoring what matters: a systematic process for selecting training-load measures. *International journal of sports physiology and performance*, 12(s2):S2–101, 2017a.

Sean Williams, Stephen West, Matthew J Cross, and Keith A Stokes. Better way to determine the acute: chronic workload ratio? *British journal of sports medicine*, 51(3):209–210, 2017b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Taedong Yun, Eric Yang, Mustafa Safdari, Jong Ha Lee, Vaishnavi Vinod Kumar, S Sara Mahdavi, Jonathan Amar, Derek Peyton, Reut Aharony, Logan Douglas Schneider, et al. Sleepless nights, sugary days: Creating synthetic users with health conditions for realistic coaching agent interactions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14159–14181, 2025.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. SOTOPIA: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*, 2023.

A ARCHETYPE SPECIFICATION

Each archetype is a YAML file. We show only the fields used in this paper; the schema also has fields for conversational simulation (onboarding persona, disclosure level, communication style) that FITSIM does not use.

```yaml
# Example: eager-beginner
#
# This listing is generated by scripts.tables.archetype_yaml.
# Source: packages/fitsim/data/archetypes/eager-beginner.yaml

simulation:
  initial_fitness: 30
  initial_fatigue: 0
  resting_hr: 75
  base_hrv: 30
  body_weight: 80
  age: 28
  sex: male
  training_age_months: 2
  fitness_gain: 0.18
  fatigue_gain: 0.12
  base_injury_risk: 4.0
  initial_motivation: 0.75
  discipline: 0.8
  initial_coach_trust: 0.5
  initial_self_efficacy: 0.65
  base_available_hours: 3.5
  work_stress_baseline: 0.3
  max_workouts_per_week: 6
  intent_days_per_week: 5
  preferred_workout_duration_min: 45
  max_workout_minutes_weekday: 60
  max_workout_minutes_weekend: 75
```

The compact table of all nine archetypes appears in Table 2 (§2.4); the per-archetype compliance decomposition follows in Appendix B.

B COMPLIANCE INTERCEPT DECOMPOSITION

Motivation, discipline, and trust are frozen at archetype-initial values throughout a benchmark run; only self-efficacy evolves daily (mastery gain $0.04(1 - \text{SE}^2)$ per workout, fatigue penalty $-0.02 \cdot \max(0, \text{fatigue} - 0.6)$). The per-archetype intercept $\beta_0(a)$ is therefore identical on every day for a given fatigue level, and $P(\text{follow})$ in the table below is the FOLLOW marginal of the joint multinomial-logit softmax (the probability the persona trains the recommended type at the recommended intensity). The complementary NATURAL marginal (same type, archetype-default intensity, no prescription bonus) and the rest probability close out the softmax. Each workout row's utility additionally subtracts $\ln R$ where $R = N + M$ (N viable types, M decoupled NATURAL rows on recommendation days); this is McFadden's standard MNL correction for the red-bus / blue-bus problem and prevents adding structurally-similar alternatives from inflating $P(\text{any workout})$.

The frozen psychological variables enter the compliance model (§2.2) through a base utility $u_{\text{base}}(a)$ that every workout row in the joint softmax shares:

$$u_{\text{base}}(a) = -3.5 + 3.5 \cdot M_a + 1.5 \cdot D_a + 0.5 \cdot S_a + 0.5 \cdot E_a + 0.5 \cdot h_a,$$

where M is motivation, D discipline, S schedule flexibility (daily-hours capacity normalized by preferred session duration), $E = 1 - \text{fatigue}$ the energy term, and $h = 1 - \exp(-w/9)$ the habit-strength term as a function of weekly compliance streak w (asymptotes over a window calibrated to the median 66-day habit-formation estimate of Lally et al. (2010); we use the median as a single deterministic timescale).

Recommendation difficulty. The recommendation's perceived *difficulty* $d \in [0.1, 0.95]$ is computed from the prescribed zone allocation rather than chosen directly. We accumulate Banister TRIMP across zones (the same kernel used for state dynamics) and compare it to the persona's typical-session TRIMP $T_{\text{typ}} = \text{CTL} \cdot 7/W_{\max}$ (where $W_{\max}$ is the archetype's max workouts per week) via a Borg-style log-sigmoid: $d_0 = \sigma\big(k \cdot \log(\text{TRIMP}/T_{\text{typ}})\big)$ with $k = 1.5$ (Borg-style log-RPE steepness). A fatigue amplifier $d = 0.5 + (d_0 - 0.5)(1 + 0.5 \cdot \text{fatigue})$ sharpens the slope at high fatigue. We then dampen d by a CTL-adaptive factor when chronic load is below 30 (an under-trained novice's perceived difficulty inflates because their typical session is small). For the tabular reference points below we use $d = 0.15$ (a constant prescription, $\approx$ the lower clamp), $d =$ archetype's initial self-efficacy (where the SE-gated refusal deviation crosses zero), and $d = 0.95$ (an all-Peak prescription, $\approx$ the upper clamp). Coach trust T enters per recommendation, not through the intercept. The FOLLOW row's full utility adds a trust-mediated prescription bonus, a self-regulation constant, and a SE-gated difficulty modifier Δ_{diff}:

$$u_{\text{follow}}(a, d) = u_{\text{base}}(a) + (0.5 + 0.5 \cdot T_a) + 0.2 + \Delta_{\text{diff}}(d, \text{SE}_a, T_a).$$

Refusal penalty. Let $\text{SE}^\star = \max(\text{SE}, d_{\text{nat}})$ floor effective self-efficacy at the persona's natural intensity (so a recommendation matching what the user does spontaneously incurs zero penalty), and define the gap $g = d - \text{SE}^\star$ and the trust-extended threshold $\theta = 0.10 + 0.10 \cdot T_a$. Then (Bandura, 1977; Locke & Latham, 2002):

$$\Delta_{\text{diff}}(d, \text{SE}, T) = \underbrace{-\frac{6.0}{4} \cdot \min\Big(\tfrac{\max(0,g)}{\theta},\, 2.0\Big)^2}_{\text{quadratic refusal}} + \underbrace{0.3 \cdot \text{SE} \cdot \big(1 - \tfrac{g}{\theta}\big) \cdot \mathbf{1}_{0<g<\theta}}_{\text{challenge bonus}}.$$

Refusal penalty reaches -1.5 logits at the threshold ($g = \theta$) and saturates at -6.0 logits when the gap reaches twice the threshold ($g = 2\theta$); the small positive challenge bonus rewards ambitious-but-attainable goals (Locke & Latham, 2002). Rest utility is $u_{\text{rest}} = -0.8 \cdot T_a$ when a recommendation exists (coaching accountability), and a fixed 0.5 logit penalty applies to any non-preferred activity type (inert under the default `cardio`-only configuration). Table 4 reports the resulting follow-row logit at neutral difficulty $d = 0.5$ (column β_0) and follow-row compliance probabilities under three difficulty levels (easy, at self-efficacy, hard).

The three P columns cover the range of recommendation difficulty: easy (all Zone-1), at-SE (the archetype's self-efficacy threshold), and hard (all Peak). The spread between $P(\text{easy})$ and $P(\text{hard})$ is

Table 4: Compliance intercept decomposition and compliance under varying recommendation difficulty. The math symbol SE here denotes self-efficacy (the user's belief in their capacity to complete a workout); it is unrelated to the archetype abbreviation SE for Social Exerciser used in the per-archetype tables. $P(\text{easy})$, $P(\text{at SE})$, and $P(\text{hard})$ are the FOLLOW marginals of the joint multinomial-logit softmax, not standalone stage probabilities, evaluated for a constant recommendation (difficulty $= 0.15$), at the archetype's self-efficacy threshold (where the linear refusal deviation crosses zero), and for an all-Peak recommendation (difficulty $= 0.95$) respectively. Day-1 conditions (fatigue $=$ initial, habit $= 0$).

Archetype	Group	M	D	S	T	SE	β_0	$P(\text{easy})$	$P(\text{at SE})$	$P(\text{hard})$
EB	Compliant	0.75	0.80	1.00	0.50	0.65	+2.28	91%	91%	2%
DDA	Compliant	0.85	0.90	1.00	0.60	0.70	+2.73	94%	94%	28%
SE	Compliant	0.75	0.55	1.00	0.60	0.50	+1.90	87%	87%	2%
IR	Constrained	0.65	0.75	1.00	0.25	0.20	-4.33	84%	84%	1%
WW	Constrained	0.40	0.35	1.00	0.45	0.45	+0.19	57%	57%	0%
SN	Constrained	0.45	0.60	1.00	0.35	0.30	-2.44	70%	70%	1%
BP	Adversarial	0.25	0.35	1.00	0.50	0.50	-0.15	46%	46%	0%
OT	Adversarial	0.75	0.85	1.00	0.40	0.80	+2.12	89%	89%	60%
SU	Adversarial	0.25	0.40	1.00	0.15	0.30	-4.81	43%	43%	0%

the coach's main lever: graded intensity is what the agent has to titrate, not just the binary choice to recommend or rest. Easy-end follow rates span 43–94% across the archetype set; hard-end follow rates collapse to 0–60% (Overtrainer is the only archetype that retains a non-trivial fraction of compliance on all-Peak prescriptions, by construction of its high self-efficacy), and the lowest-motivation archetypes lose the most compliance to intensity in relative terms (Silent User and Busy Professional drop to near-zero).

C ORACLE REFERENCE TRAJECTORY

The oracle is a DP baseline with hidden-state access. It reads the simulator's full hidden state (Banister Δf and fatigue, injury history, exponentially-weighted moving-average (EWMA) load) and imports the simulator's exact Banister and compliance constants. Its plans exploit both: perfect state observation *and* perfect parametric form.

The oracle is not provably optimal: compliance is nonlinear, injury probability is path-dependent, and the 3-iteration DP is not guaranteed to find the global optimum. It is also not a coaching ceiling: deterministic reference coaches can beat it on specific (archetype, seed) cells where the DP's plan loses to a fixed rule under the compliance draw (Reference-trajectory integrity table below). The LLM-to-oracle gap therefore bundles two costs — hidden-state inference and parametric-form rederivation — and we treat the DP as a diagnostic baseline, not a ceiling. Disentangling the two costs (e.g., observation-only oracle, oracle with perturbed parameters) is future work.

Formulation. The oracle parameterizes a recommendation by a scalar intensity $\lambda \in [0, 1]$ that linearly interpolates zone allocation: $\lambda = 0$ assigns all minutes to OUT OF RANGE, $\lambda = 1/3$ to FAT BURN, $\lambda = 2/3$ to CARDIO, $\lambda = 1$ to PEAK, with intermediate values splitting between adjacent zones (the final zone absorbs minute-rounding remainders). The DP enumerates $\lambda \in \{0.0, 0.1, \ldots, 1.0\}$ and solves a day-by-day optimization:

$$\text{netGain}(\lambda) = \Delta G(\lambda) - C_{\text{fatigue}}(\lambda) - C_{\text{recovery}}(\lambda) - P_{\text{inj}}(\lambda, s) \cdot OC_d$$

Expected fitness gain (adaptation-saturation heuristic, consistent with Busso (Busso, 2003)):

$$\Delta G = K_1 \cdot \text{TRIMP}(\lambda) \cdot g_f \cdot (1 - 0.3 \cdot \text{fatigue}) \cdot e^{-\alpha \cdot \max(0, \Delta f)}$$

where g_f is the archetype's fitness gain rate and K_1 is the Banister fitness-adaptation constant.

Fatigue opportunity cost (multi-day projection):

$$C_{\text{fatigue}} = 0.3 \cdot \Delta G \cdot \frac{K_2 \cdot \text{TRIMP} \cdot g_{\text{fat}}}{100} \cdot \tau_2(1 - e^{-3}) \cdot \frac{f_{\text{intent}}}{7}$$

where g_{fat} is the archetype's fatigue gain rate (`fatigue_gain` in the YAML schema) and K_2 is the Banister fatigue-decay constant.

The 0.3 coefficient derives from the diminishing factor $\text{dim} = 1 - 0.3 \cdot \text{fatigue}$. The horizon sum captures 95% of fatigue decay. Frequency scaling $f_{\text{intent}}/7$ avoids penalizing rest days.

Intensity recovery cost (following Foster (Foster, 1998), Seiler (Seiler, 2010)):

$$C_{\text{recovery}} = C_r \cdot \lambda^2 \cdot \Delta G$$

where $C_r = 0.3$ (calibrated). The λ^2 shape reflects that high-intensity sessions require disproportionately more recovery, creating an interior optimum around $\lambda = 0.4$–0.6 consistent with polarized training (Seiler & Kjerland, 2006; Stöggl & Sperlich, 2015).

Injury opportunity cost (DP backward pass):

$$OC_d = \sum_{i=d+1}^{d+R} \text{gain}_i + \max(0, \Delta f_d) \cdot (1 - e^{-R/\tau_1})$$

where $R = 23$ days (weighted mean from injury registry). The first term sums missed gains during recovery; the second captures detraining loss. Injury probability is computed from projected ACWR (Gabbett, 2016).

Two-pass DP. The oracle iterates: (1) forward simulate with best-λ sweep per intent day, (2) backward-compute OC from forward gains, (3) re-optimize with OC-informed injury costs. Converges in 2–3 iterations.

Compliance-aware oracle. The informed oracle weights gain by compliance:

$$\mathbb{E}[\text{TRIMP}] = P_{\text{type}} \cdot \text{TRIMP}_{\text{rx}} + P_{\text{sub}} \cdot \text{TRIMP}_{\text{natural}}$$

P_{type} is the FOLLOW marginal of the joint multinomial-logit softmax (the probability the persona trains the recommended type at the recommended intensity); P_{sub} is the NATURAL marginal (substitutes natural-intensity training of the same type, no prescription bonus). In the cardio-only action space, on recommendation days the softmax has three rows — {follow recommendation, natural-intensity cardio, rest} — and the oracle weights both training rows separately. On intent days without an active recommendation, only {natural-intensity cardio, rest} are scored.

Limitations. The oracle knows the exact compliance function and injury probability model that a real coach would have to estimate. Psychology is frozen, so coaching quality cannot improve compliance. Injuries in benchmark mode are deterministic ACWR-threshold events, not stochastic rolls. The oracle has no rest, recovery, communication, or motivational channel beyond load. The DP heuristic is not provably optimal (nonlinear compliance, path-dependent ACWR).

Results. The informed oracle achieves an aggregate CGR of 1.44 (95% CI $[1.35, 1.55]$, geometric mean across 9 archetypes, 5 seeds, 365-day horizon) against the null-coach baseline (full row in Table 3). The oracle matches or exceeds the baseline on eight of the nine archetypes; for Overtrainer the oracle CGR sits at 0.93 (95% CI [0.91, 0.95]). This is by construction of the compliance-conservative DP facing a high-fatigue archetype: the natural-intensity routine in the baseline accumulates gains that the oracle's safety gates decline.

C.1 PARTIALLY-OBSERVED ORACLE

The informed oracle bundles two costs together — hidden-state inference and parametric-form rederivation. To disentangle them we add a **partially-observed oracle**: the same DP planner, but with hidden state replaced by reconstructions of the same observation surface the LLM agent sees through the mock API (sleep, RHR, HRV, activity, VO2max, body weight) and via `list_workouts`. It does not read hidden Banister Δf or fatigue, ACWR, compliance coefficients, injury parameters, or any archetype field that is not exposed to the LLM through the system prompt or a tool output.

Demographics from the system prompt. The same fields the LLM receives in its prompt are passed through to the synthetic archetype fed into the planner: `age`, `sex`, `max_workout_minutes_weekday`, and `max_workout_minutes_weekend`. Body weight is read from the `get_weight` tool output rather than hardcoded. Hidden archetype fields (fitness/fatigue gains, injury risk, discipline, motivation priors) stay at their population-mean defaults. Without this passthrough the planner runs every persona through a 35-year-old female with the population-mean duration caps, which compresses HR-zone calculations and the compliance softmax.

State reconstruction from the workout database and biometrics. The simulator's hidden-state dynamics (ATL/CTL accumulation, Δf update, fatigue accrual) are driven by Banister TRIMP, computed from each executed segment's average HR. The partial oracle does not see average HR per workout; only the per-zone time-in-zone allocation that an LLM coach would read from a workout payload. We reconstruct Banister TRIMP from those zone allocations: the minutes-weighted sum of zone-centre $\%\text{HRR}$ midpoints (OOR 0.30, FAT BURN 0.60, CARDIO 0.78, PEAK 0.93) gives an estimated average $\%\text{HRR}$, which combines with the persona's Tanaka max HR (from age) and the latest observed RHR to yield an $\overline{\text{HR}}$. That feeds the same $\text{TRIMP} = \text{duration} \cdot \Delta\text{HR} \cdot b \cdot \exp(y \cdot \Delta\text{HR})$ formula the engine uses, with the persona's sex-specific b / y coefficients. Because the Banister kernel is convex in ΔHR, evaluating it at the minutes-weighted average HRR rather than averaging segment-level TRIMP is a Jensen-inequality lower bound: $\text{TRIMP}(\mathbb{E}[\Delta\text{HR}]) \leq \mathbb{E}[\text{TRIMP}(\Delta\text{HR})]$. The downward bias is small in practice because HR zone bins are narrow and within-zone HR drift is a second-order effect, but it pushes the partial oracle's ATL/CTL estimates conservative on workouts that mix CARDIO and PEAK time. ATL/CTL/ACWR then accumulate this reconstructed TRIMP via the same EWMA the engine uses for hidden-state ATL/CTL. Δf rolls forward via the published Banister recursion seeded at zero, with the *fitness baseline* anchored from the trailing-window VO2max observations (inverting the simulator's emission curve $\text{VO2} \approx 20 + (\text{fitness}/100) \cdot 40$, falling back to the population mean before five observations have accumulated). Compliance

probability is the rolling 14-day completion rate (initialized at 0.7); the per-archetype gain rate g_f defaults to 0.20 with an online clamp to $[0.10, 0.30]$, refined after day 21 by an OLS regression of VO2max on cumulative TRIMP. Intent days, the preferred workout duration, and the weekly workout cap are inferred from the rolling completion record — the LLM sees the same statistics through `list_workouts`. Consecutive high-fatigue days, which feeds the planner's overtraining-suppression term, are reconstructed as the trailing streak above the engine's 0.8 fatigue threshold using the posterior fatigue defined below.

Hybrid fatigue posterior. The planner consumes a fatigue posterior produced by inverse-variance weighting three observation channels — RHR (with a $+22\%$-at-max amplification, an authorial calibration choice within the qualitative magnitude range associated with overreaching by Halson & Jeukendrup (2004)), HRV (with a -40%-at-max suppression, an authorial calibration choice within the qualitative RMSSD response range reviewed by Plews et al. (2013)), and sleep efficiency. The fused observation is then α-blended against the Banister-rolled prior with a cold-start ramp ($\alpha = 1.0$ during the first 14 days, decaying linearly to $\alpha = 0.6$ by day 21) so the prior stays dominant at steady state. The $+22\%/-40\%$ coefficients match the simulator's biometric emission layer (Appendix D.2); the filter inverts the data-generating curve rather than introducing a new one. This shared parametric form is one reason we read the partial oracle as a *ceiling under shared parametric form*, not a pure observability ceiling — an LLM coach without that form would face a larger inverse-modelling cost.

Results. Table 3 reports the full configuration, which combines all reconstruction components above: demographics passthrough, history-inferred preferred duration and weekly cap, the hybrid Bayesian fatigue posterior, VO2-anchored fitness baseline, observed-zone TRIMP, the observed-natural-zone preference (rolling 60-prescription histogram of completed-workout zone allocations, including substitutes that didn't match the prescription), a self-efficacy proxy from rolling completion-at-high-λ ($\lambda \geq 0.6$), a 28-day rolling OLS regression of VO2max on cumulative TRIMP that estimates g_f after day 21, and a reconstructed `consecutive_high_fatigue_days` streak that re-enables the planner's overtraining-suppression term (CGR 1.21, 95% CI $[1.12, 1.32]$, InjD 23.0). Per-archetype the partial oracle clears 1.0 on every cell (CGR 1.07–1.39 point estimates), but the gap to the informed oracle opens up wherever the partial proxies cannot recover the hidden state cleanly: Silent User ($2.20 \rightarrow 1.27$) and Busy Professional ($2.09 \rightarrow 1.24$) lose the most because their sparse compliance yields too few proxy updates, while Weekend Warrior ($1.46 \rightarrow 1.20$), Injury Returnee ($1.27 \rightarrow 1.19$), and Data-Driven Athlete ($1.22 \rightarrow 1.14$) lose less but still leave coaching headroom on the table.

Decomposition. The partial oracle quantifies the *shared-parametric-form ceiling under partial observation*: informed $\rightarrow$ partial costs 0.23 CGR ($1.44 \rightarrow 1.21$), the price of inferring fitness/fatigue/ACWR from biometrics rather than reading them directly while holding the simulator's parametric form fixed. The partial oracle still uses the simulator's parametric form, so 0.23 is a lower bound on what an LLM without that form would face on observation alone. The remaining partial-to-LLM gap and its within-family scaling behaviour are discussed in §4. We read the partial-oracle CGR as a coachability signal: archetypes whose partially-observed CGR clears 1.0 are genuinely coachable from the wearable observation surface alone, so their LLM-to-partial gap is recoverable in principle through better prompting and tool use; archetypes below 1.0 require either richer observability (e.g., self-report, calendar context) or a different coaching objective. All nine archetypes clear 1.0 in the partial-oracle point estimate in our grid (range 1.07–1.39); Busy Professional, Silent User, Eager Beginner, and Weekend Warrior have lower 95% CI bounds dipping below 1.0, and Skeptical Newcomer is the strongest (1.39 [1.29, 1.59]).

Reference-trajectory integrity. Table 5 reports per-archetype oracle CEI together with the number of (archetype, seed) cells on which any deterministic reference coach (No coaching, Constant, Linear deload) equalled or exceeded the oracle. Such exceedances are expected at the tails of low-compliance archetypes under the compliance-weighted objective — when the DP's optimal-under-compliance recommendation is conservative enough that a fixed rule occasionally matches or beats it on a specific seed — and are logged for transparency by the oracle-ceiling audit. Across our grid, 11 of 135 (archetype, seed, baseline) cells exceed the oracle, concentrated on Overtrainer (10/15, 66.7%) and Weekend Warrior (1/15, 6.7%), with zero exceedances across all other seven archetypes. The Overtrainer concentration is consistent with the compliance-aware DP being conservative against a high-fatigue archetype: the natural-intensity training pattern in the null and fixed-rule baselines occasionally accumulates fitness gain that the oracle's compliance-conservative recommendations decline. A high exceedance rate for a given archetype signals DP-heuristic shortcomings that future iterations should narrow, not a broken metric.

Table 5: Oracle CEI and deterministic-baseline exceedance rate by archetype (5 seeds × 3 reference baselines = 15 cells per archetype; seed resampling yields the bootstrap CIs). Oracle CEI is reported in raw CEI units (fitness delta per $\sqrt{E_h} \cdot g_f$) — to convert to CGR divide by the null-coach CEI on the same archetype (Table 6). An exceedance is a (seed, reference coach) pair whose CEI equals or exceeds the oracle's for that seed — expected at the tails of low-compliance archetypes because the oracle is a DP baseline with direct hidden-state access, not an achievable ceiling.

Archetype	Oracle CEI (mean)	Exceedances	Exceedance rate
eager-beginner	9.106	0	0.0%
data-driven-athlete	10.721	0	0.0%
social-exerciser	11.786	0	0.0%
busy-professional	6.975	0	0.0%
injury-returnee	9.904	0	0.0%
weekend-warrior	6.072	1	6.7%
skeptical-newcomer	8.712	0	0.0%
overtrainer	7.085	10	66.7%
silent-user	5.174	0	0.0%

D Observation Function $Z(o \mid s)$

The coaching agent observes the hidden state exclusively through mock API tool responses. All observations are **date-deterministic**: a seeded hash $\text{dateVar}(d, k) \in [-1, 1)$ (xorshift32 on day number and seed k) ensures reproducibility while introducing realistic day-to-day variation.

D.1 Biometrics (intermediate values from hidden state)

$$\text{RHR} = \big(\text{RHR}_0 - 0.6 \cdot (1 - (f/100)^2) \cdot \Delta f\big) \cdot (1 + 0.22 \cdot \text{fatigue})$$

$$\text{HRV} = \max\big(0,\ (\text{HRV}_0 + 0.3\,\Delta f) \cdot (1 - 0.4 \cdot \text{fatigue}) \cdot \text{SQ}\big)$$

$$\text{SQ} = \text{clamp}\big(\text{SQ}_0 - 0.20 \cdot \text{stress} - 0.15 \cdot \text{fatigue} + \varepsilon,\ 0.1,\ 1.0\big)$$

where f = fitness level, Δf = fitness delta, SQ = sleep quality (dimensionless, $[0.1, 1.0]$), and $\varepsilon \sim \mathcal{N}(0, 0.05)$. RHR coefficients are calibrated within the trained-vs-untrained ranges discussed by Reimers et al. (2018); the $+22\%$-at-max fatigue amplification on RHR is an authorial calibration choice within the qualitative magnitude range associated with overreaching by Halson & Jeukendrup (2004); and the -40%-at-max HRV suppression is an authorial calibration choice within the qualitative RMSSD response range reviewed by Plews et al. (2013).

D.2 Per-Tool Observation Functions

Each observation tool below is listed with (a) a plain-language description, (b) the LLM-visible input schema, (c) the noise model that maps hidden state to the observed values, and (d) a sample of the text the coach receives. All 9 tools share the same date-range schema: two optional `YYYY-MM-DD` parameters that default to a trailing 7-day window if omitted. Outputs are factual data tables without editorial prose — no "narrow range" hints, no "Best/Worst Day" footers, no goals rows, no zone-time summaries, no lower-RHR hints, no sedentary warnings, no single-row averages beyond the table itself. The agent does the reasoning; the tool does not pre-compute advice.

`fitbit_get_sleep` Sleep stages, duration, efficiency, and onset latency for each night in a date range. A coach calls this to read overnight recovery quality, identify cumulative sleep debt, and contextualise HRV / RHR readings against sleep architecture.

Input:

- `start_date`: `YYYY-MM-DD` (optional; defaults to 7 days before `end_date`).

- `end_date`: `YYYY-MM-DD` (optional; defaults to today).

Noise model. TST is total sleep time and SOL is sleep onset latency.

$$\text{TST} = \text{clamp}\big((270 + \text{SQ} \cdot 210) \cdot (1 + 0.15 \cdot \text{dateVar}),\ 240,\ 600\big)\ \text{min}$$

$$\text{SOL} \sim \text{LogNormal}\big(\text{clamp}(9 + 3\,(1 - \text{SQ}) + 8\,\text{stress},\ 5,\ 30),\ \text{cv} = 0.4\big)\ \text{min}$$

Stages: deep $= 0.18 \cdot \text{TST} + \mathcal{N}(0, 10)$, REM $= 0.22 \cdot \text{TST} + \mathcal{N}(0, 10)$, light = remainder.

Output:

```
[Data context] User's Fitbit sleep data. "Main sleep" is the
longest period; shorter periods are naps. Data reflects the night
BEFORE each date.

Sleep Data (Apr 10, 2026 - Apr 12, 2026):

| Date | Duration | Deep | REM | WASO | Latency | Awaken |
|---|---|---|---|---|---|---|
| Apr 12, 2026 | 7h 08m | 78m | 95m | 38m | 11m | 5 |
| Apr 11, 2026 | 6h 52m | 71m | 89m | 41m | 14m | 6 |
| Apr 10, 2026 | 7h 24m | 84m | 102m | 33m | 9m | 4 |
| **Average** | 7h 08m | 78m | 95m | 37m | 11m | 5 |
```

`fitbit_get_heart_rate` Per-day resting HR plus minutes spent in each HR zone (Out of Range, Fat Burn, Cardio, Peak). A coach uses this to track cardiovascular adaptation (lower RHR over weeks) and to verify the intensity distribution of completed sessions.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional). FITSIM caps the agent at one call per day per tool, so observations are aggregated over the requested range rather than queried at intraday resolution.

Noise model:

$$\text{RHR}_{\text{obs}} = \text{round}\big(\mathcal{N}(\text{RHR},\ \sigma{=}1.5)\big), \quad \text{clamped } [38, 110]$$

Zone boundaries from Tanaka: $\text{HR}_{\max} = 208 - 0.7 \cdot \text{age}$, with zone splits at 50%, 70%, 85% of max.

Output:

```
[Data context] User's Fitbit heart rate data. Resting HR is
derived from overnight sleep data and periods of daytime
inactivity -- it does not update during the day. Zone minutes
are cumulative and update throughout the day. Weekly RHR trends
are more meaningful than single days.

Heart Rate Data (Apr 06, 2026 - Apr 12, 2026):

| Date | Resting HR | OoR | Fat Burn | Cardio | Peak |
|---|---|---|---|---|---|
| Apr 12, 2026 | 62 bpm | 1378 min | 12 min | 18 min | 4 min |
| Apr 11, 2026 | 61 bpm | 1440 min | 0 min | 0 min | 0 min |
| ...
Average Resting HR: 62 bpm
```

`fitbit_get_hrv` Per-night daily-RMSSD and deep-sleep-RMSSD. Coaches compare against the user's own 7–14 day baseline to gauge autonomic recovery; a single night below baseline after a hard session is normal adaptation, not a red flag.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Noise model: $\text{RMSSD}_{\text{obs}} \sim \text{LogNormal}(\text{HRV},\ \text{cv}{=}0.15)$ ms.

Output:

```
[Data context] User's Fitbit HRV data. HRV (RMSSD) is measured
exclusively during sleep -- each date's value reflects the
preceding night.

HRV Data (Apr 06, 2026 - Apr 12, 2026):

| Date | Daily RMSSD | Deep Sleep RMSSD |
|---|---|---|
| Apr 12, 2026 | 42.3 ms | 47.1 ms |
| Apr 11, 2026 | 39.8 ms | 44.5 ms |
| ...
| **Average** | 41.2 ms | 45.8 ms |
```

`fitbit_get_activity` Daily steps, distance, active-minute breakdown (light/fairly/very), sedentary minutes, total and BMR calories, activity calories, and floors. Used to read whether the user moves outside scheduled sessions and to contextualise recommended-vs.-actual training volume.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Noise model:

$$\text{steps} \sim \mathcal{N}\big(5000 + 3000 \cdot M + b_{\text{workout}},\ \text{cv}{=}0.25\big)$$

where $b_{\text{workout}} = 3000$ if a workout occurred, M = motivation (frozen). Weekend adjustment: -1200 steps.

Output:

```
[Data context] User's Fitbit activity data. Steps, active
minutes, and calories are cumulative throughout the day and only
complete after midnight.

Activity (Apr 06, 2026 - Apr 12, 2026):

| Date | Steps | Dist | Active Min (L/F/V) | Sed Min | Act Cal | Floors |
|---|---|---|---|---|---|---|
| Apr 12, 2026 | 8,432 | 6.1km | 84 (48/22/14) | 1,356 | 410 | 6 |
| Apr 11, 2026 | 7,118 | 5.2km | 62 (40/15/7) | 1,378 | 280 | 4 |
| ...
| **Average** | 7,856 | 5.7km | 71 (43/18/10) | 1,369 | 332 | 5 |
```

`fitbit_get_steps_trend` Daily step counts over a date range with comparisons against the user's step goal and the period average. More efficient than `fitbit_get_activity` for long ranges (uses time-series API rather than per-day fetches). A coach uses this to spot longitudinal step trends.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Output:

```
[Data context] User's Fitbit steps data. Daily totals may include
off-wrist steps (phone in pocket/bag) which intraday data does not.

Steps Trend (Apr 06, 2026 - Apr 12, 2026):

| Date | Steps | vs Goal | vs Avg |
|---|---|---|---|
| Apr 12, 2026 | 8,432 | 84% | +7% |
| Apr 11, 2026 | 7,118 | 71% | -9% |
| ...
Period Average: 7,856 steps/day
Days Meeting Goal: 2/7
```

`fitbit_get_sedentary_time` Per-day breakdown of sedentary, lightly active, fairly active, and very active minutes. A coach reading this checks whether sedentary time is displacing rec-

ommended dose — a high-sedentary user with a single hard session may still under-train weekly volume.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Output:

```
[Data context] User's Fitbit activity data. Activity level
classification (sedentary/light/moderate/vigorous) is based on
movement intensity.

Activity Breakdown (Apr 06, 2026 - Apr 12, 2026):

| Date | Sedentary | Light | Moderate | Vigorous |
|---|---|---|---|---|
| Apr 12, 2026 | 1356 min | 48 min | 22 min | 14 min |
| ...
Daily Averages:
* Sedentary: 1380 min/day (23.0 hours)
* Active: 71 min/day (light: 43, moderate: 18, vigorous: 10)
```

`fitbit_get_weight` Weight log entries (kg, BMI, optional body fat %, source) over a date range. Not consumed by any archetype's state dynamic — retained in the tool inventory for realism. A coach may call this opportunistically; the simulator returns no entries unless a fixture seeds them.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Output (typical empty case):

```
No weight data available for Apr 06, 2026 to Apr 12, 2026.
Weight data requires manual entry or a connected smart scale.
```

`fitbit_get_skin_temperature` Per-night relative skin-temperature deviation from the user's 30-night baseline (positive = warmer than baseline). A coach treats sustained $> +1.0\,^\circ$ as a possible illness or overtraining signal.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Noise model:

$$\Delta T_{\text{skin}} = \max(0,\ 0.6 \cdot (\text{fatigue} - 0.5)) + \mathcal{N}(0, 0.15)\ ^\circ\text{C}$$

Output:

```
[Data context] User's Fitbit skin temperature data. Measured
during sleep only.

Skin Temperature (Apr 06, 2026 - Apr 12, 2026):

| Date | Variation | Sensor |
|---|---|---|
| Apr 12, 2026 | +0.32 | nightly |
| Apr 11, 2026 | +0.41 | nightly |
| ...
Average Variation: +0.28
Range: -0.05 to +0.41
```

`fitbit_get_vo2_max` Daily Cardio Fitness Score (VO_2 max range, ml/kg/min). A coach uses this as a slow-moving aerobic-fitness anchor; it is a deterministic function of Banister fitness in the simulator.

Input: `start_date`, `end_date` (`YYYY-MM-DD`, both optional).

Noise model:

$$\text{VO}_2\text{max} = 20 + 0.4 \cdot f + \mathcal{N}(0, 1), \qquad f \in [0, 100]$$

Output:

```
[Data context] User's Fitbit VO2 Max data. VO2 Max updates only
after aerobic exercise of sufficient intensity -- missing days
are expected, not data gaps.

VO2 Max Trend (Mar 14, 2026 - Apr 12, 2026):

| Date | VO2 Max Range | Midpoint |
|---|---|---|
| Apr 09, 2026 | 44-48 | 46.0 ml/kg/min |
| Apr 02, 2026 | 43-47 | 45.0 ml/kg/min |
| ...
Current: 44-48 ml/kg/min
Period Average: 45.4 ml/kg/min
```

Ablated in FITSIM: `fitbit_get_health_metrics` The API exposes a composite snapshot aggregating RHR, HRV, breathing rate ($20 - 8 \cdot \mathrm{SQ} \pm 0.5$), SpO_2 ($97\% \pm 1\%$), and skin temperature deviation, with internal z-scores and significance labels over a 14-day rolling window. The default FITSIM tool surface omits this endpoint so the coach must compute its own trend statistics from the per-metric tools above — a capability we want CGR to score, not hand to the model. We re-enable it in the §G ablation to measure whether providing pre-computed signals improves recommendation quality.

Output (abridged JSON, one entry per metric):

```
{
  "hrv_rmssd": {"window_mean": 41.2, "window_stdev": 3.6,
                "recent_mean": 39.7, "z_score": -0.42,
                "trend_slope_per_day": -0.08,
                "recent_values": [42.3, 39.8, ...]},
  "resting_hr": {...},
  "sleep_efficiency": {...},
  "daily_steps": {...},
  "body_weight_kg": {...},
  "vo2_max": {...}
}
```

D.3 WORKOUT-HISTORY TOOLS

Two read-only tools let the agent inspect its own past recommendations: `list_workouts` and `get_workout`. They read from the workout database that the recommendation tools write to (Appendix D.4), so the agent can recall the rationale and zone targets it itself authored on a previous day. Both return JSON records with `id`, `title`, `date`, `status`, `metadata`, and `created_at` fields — the activity type is implicit (always `cardio`) and not part of the returned projection.

`list_workouts` List coach-recommended workouts. Returns a JSON array of {`id`, `title`, `date`, `status`, `metadata`, `created_at`} records.

Input:

- `status`: `pending` / `completed` / `skipped` (optional filter).
- `limit`: integer 1–50 (optional; default 10).
- `direction`: `recent` / `upcoming` / `all` (optional; default `recent`).

Output:

```
[Data context] Workouts you proposed for the user. Status
reflects whether the user completed, skipped, or hasn't yet
attempted each proposal.

[
  {
    "id": "7f3b9c2a-...",
```

```
    "title": "Easy Zone-2",
    "date": "2026-04-13",
    "status": "completed",
    "metadata": {
      "duration": 45,
      "target_hr_zones": [
        {"zone": "Fat Burn", "minutes": 15},
        {"zone": "Cardio", "minutes": 25},
        {"zone": "Peak", "minutes": 5}
      ]
    },
    "created_at": "2026-04-13T07:30:00.000Z"
  }
]
```


`get_workout` Fetch the full content of a single workout by ID prefix. Used to read back rich `details` / `description` markdown the coach itself authored.

Input: `workout_id` (string ≥ 8 chars).

Output:

```
[Data context] Workout you proposed for the user.

## Easy Zone-2
Date: 2026-04-13 | Status: completed
Duration: 45 min
ID: 7f3b9c2a-...

### Description
Aerobic base session targeting Fat Burn / low Cardio.

### Workout Details
```markdown
- 10 min warm-up, RPE 3
- 30 min steady, target 70-78% HRmax
- 5 min cool-down
```
```

D.4 Recommendation Tools

FitSim exposes two write tools, `create_workout` and `update_workout`. Both take a single `date` (no date range) and fix the activity type to `cardio`, so the agent's choice is reduced to duration and HR-zone distribution.

`create_workout` Create a cardio recommendation for a specific date. The benchmark scores the duration $\times$ HR-zone-distribution choice; the zone-minutes total must equal `duration` within ± 5 min.

Input:

- `title`: string (required) — e.g., "Easy Zone-2", "Threshold Intervals".
- `date`: `YYYY-MM-DD` (required).
- `duration`: number, 5–240 (optional; defaults to 45 if omitted).
- `target_hr_zones`: array of `{zone, minutes}` (optional but expected; the system prompt makes it mandatory). `zone` is one of `Out of Range`, `Fat Burn`, `Cardio`, `Peak`.
- `subtitle`, `description`, `details`: optional strings (free-form prose). These freeform fields are designed as a primitive form of agentic memory: the agent can annotate its rationale on day t and revisit it when revising the recommendation on day $t+1$ (the same fields are returned by `list_workouts` / `get_workout`).

Output (success):

```
Successfully created cardio workout "Easy Zone-2" scheduled for
2026-04-13.
ID: 7f3b9c2a-...
```

Output (validation failure example, e.g. duration $= 45$*, zone-minutes sum* $= 30$*):*

```
Error: target_hr_zones total (30 min) must equal duration
(45 min) within +/- 5 min. Adjust zone minutes to sum to the
workout duration.
```

`update_workout` Modify an existing workout — reschedule (`date`), mark `completed` / `skipped`, or revise content. The activity type is fixed and cannot be changed.

Input:

- `workout_id`: string ≥ 8 chars (UUID prefix or full UUID).
- Optional updates: `title`, `subtitle`, `description`, `details`, `date` (`YYYY-MM-DD`), `status` (`pending` / `completed` / `skipped`), `duration` (5–240), `target_hr_zones` (same shape as in `create_workout`).

Output (success):

```
Successfully updated workout "Easy Zone-2". Changes:
status: "completed", scheduled_date: "2026-04-13"
```

E PER-ARCHETYPE RESULTS AND SAFETY

Table 6 reports per-archetype CGR, safety diagnostics (injury days, ACWR violations, overtraining episodes), and two orthogonal compliance metrics that decompose training volume into "did the coach engage?" and "did the user listen?": **Rx** is the fraction of simulation days the coach issued a recommendation (vs. explicitly choosing rest), and **Follow** is, of those recommended days, the fraction where the user completed the recommended workout. Overall training volume on intent days is approximately Rx $\times$ Follow.

Table 6: Per-archetype breakdown (365 days, 5 seeds). Archetypes: EB Eager Beginner, DDA Data-Driven Athlete, SE Social Exerciser, BP Busy Professional, IR Injury Returnee, WW Weekend Warrior, SN Skeptical Newcomer, OT Overtrainer, SU Silent User. Columns: InjD mean injury days; ACWR mean days with acute:chronic workload ratio > 1.3 (Gabbett sweet-spot edge, matching Table 3); OT Ep. mean count of high-fatigue streaks (fatigue > 0.6 for > 5 consecutive days); Rx fraction of days the coach recommended rather than rested; Follow fraction of recommended days the user completed. Follow is reported as "—" for No coaching (no recommendations issued).

Model	Arch.	CGR	InjD	ACWR	OT Ep.	Rx	Follow
Null coach	BP	1.00 [1.00, 1.00]	0.0	65.4	0.00	0.00	—
Null coach	DDA	1.00 [1.00, 1.00]	0.0	11.4	0.00	0.00	—
Null coach	EB	1.00 [1.00, 1.00]	0.0	18.8	0.00	0.00	—
Null coach	IR	1.00 [1.00, 1.00]	4.6	44.6	0.00	0.00	—
Null coach	OT	1.00 [1.00, 1.00]	13.8	18.2	3.80	0.00	—
Null coach	SU	1.00 [1.00, 1.00]	0.0	14.2	0.00	0.00	—
Null coach	SN	1.00 [1.00, 1.00]	0.0	63.2	0.00	0.00	—
Null coach	SE	1.00 [1.00, 1.00]	0.0	52.0	0.00	0.00	—
Null coach	WW	1.00 [1.00, 1.00]	4.6	76.2	0.00	0.00	—
Constant Zone-1	BP	1.47 [1.22, 1.86]	0.0	60.2	0.00	0.43	0.41
Constant Zone-1	DDA	0.81 [0.77, 0.85]	4.6	38.6	0.00	0.72	0.94
Constant Zone-1	EB	0.91 [0.87, 0.94]	0.0	13.0	0.00	0.72	0.92
Constant Zone-1	IR	0.97 [0.92, 1.00]	0.0	27.8	0.00	0.57	0.83
Constant Zone-1	OT	0.86 [0.81, 0.91]	9.2	28.4	0.00	0.72	0.88
Constant Zone-1	SU	0.97 [0.66, 1.62]	0.0	11.2	0.00	0.29	0.30
Constant Zone-1	SN	1.17 [1.00, 1.38]	0.0	60.8	0.00	0.29	0.64
Constant Zone-1	SE	0.88 [0.81, 0.98]	0.0	43.0	0.00	0.43	0.89
Constant Zone-1	WW	0.91 [0.79, 1.08]	0.0	54.6	0.00	0.29	0.52
Linear-deload	BP	1.58 [1.20, 2.13]	0.0	79.8	0.00	0.43	0.40
Linear-deload	DDA	0.95 [0.93, 0.96]	9.2	20.6	0.00	0.64	0.92
Linear-deload	EB	1.04 [1.01, 1.08]	0.0	10.8	0.00	0.64	0.93
Linear-deload	IR	1.08 [1.06, 1.12]	0.0	37.0	0.00	0.50	0.81
Linear-deload	OT	0.98 [0.95, 1.00]	9.2	11.0	1.40	0.64	0.88
Linear-deload	SU	1.34 [0.93, 2.31]	0.0	33.8	0.00	0.25	0.30
Linear-deload	SN	1.33 [1.18, 1.53]	0.0	79.6	0.00	0.25	0.60
Linear-deload	SE	0.94 [0.82, 1.08]	0.0	27.8	0.00	0.43	0.89
Linear-deload	WW	1.15 [0.99, 1.35]	0.0	76.2	0.00	0.25	0.48
Oracle (informed)	BP	2.09 [1.61, 2.86]	4.6	89.2	0.00	0.43	0.42
Oracle (informed)	DDA	1.22 [1.21, 1.23]	0.0	14.6	8.40	0.72	0.94
Oracle (informed)	EB	1.21 [1.19, 1.22]	4.6	11.2	0.40	0.71	0.91
Oracle (informed)	IR	1.27 [1.21, 1.32]	0.0	20.6	0.00	0.57	0.80
Oracle (informed)	OT	0.93 [0.91, 0.95]	0.0	13.4	0.80	0.72	0.90
Oracle (informed)	SU	2.20 [1.46, 3.83]	0.0	59.4	0.00	0.29	0.31
Oracle (informed)	SN	1.65 [1.39, 1.97]	0.0	88.0	0.00	0.29	0.62
Oracle (informed)	SE	1.34 [1.22, 1.51]	0.0	35.4	0.20	0.43	0.88
Oracle (informed)	WW	1.46 [1.21, 1.80]	4.6	92.4	0.00	0.28	0.53
Partial oracle	BP	1.24 [0.89, 1.81]	0.0	83.8	0.00	0.16	0.34
Partial oracle	DDA	1.14 [1.10, 1.19]	0.0	26.8	7.20	0.29	0.95
Partial oracle	EB	1.07 [0.99, 1.19]	4.6	27.2	1.60	0.27	0.92

(continued on next page)

Table 6 continued from previous page

Model	Arch.	CGR	InjD	ACWR	OT Ep.	Rx	Follow
Partial oracle	IR	1.19 [1.13, 1.25]	0.0	48.6	0.00	0.19	0.80
Partial oracle	OT	1.17 [1.06, 1.27]	18.4	37.2	19.80	0.24	0.87
Partial oracle	SU	1.27 [0.80, 2.38]	0.0	31.2	0.00	0.16	0.30
Partial oracle	SN	1.39 [1.29, 1.59]	0.0	80.8	0.00	0.16	0.58
Partial oracle	SE	1.25 [1.15, 1.38]	0.0	57.8	0.00	0.16	0.80
Partial oracle	WW	1.20 [0.96, 1.52]	0.0	84.8	0.00	0.16	0.47
Gemini 3.1 Flash-Lite	BP	1.46 [1.05, 2.11]	0.0	70.6	0.00	0.41	0.15
Gemini 3.1 Flash-Lite	DDA	0.99 [0.96, 1.01]	0.0	13.4	0.00	0.44	0.71
Gemini 3.1 Flash-Lite	EB	1.02 [0.99, 1.07]	0.0	14.0	0.00	0.39	0.71
Gemini 3.1 Flash-Lite	IR	1.01 [0.98, 1.04]	0.0	33.6	0.00	0.44	0.51
Gemini 3.1 Flash-Lite	OT	0.98 [0.97, 1.00]	0.0	11.2	2.00	0.43	0.66
Gemini 3.1 Flash-Lite	SU	0.75 [0.46, 1.12]	0.0	11.6	0.00	0.27	0.08
Gemini 3.1 Flash-Lite	SN	1.05 [0.96, 1.16]	0.0	63.2	0.00	0.33	0.15
Gemini 3.1 Flash-Lite	SE	1.07 [0.97, 1.21]	0.0	34.0	0.00	0.39	0.41
Gemini 3.1 Flash-Lite	WW	1.19 [0.96, 1.49]	0.0	78.8	0.00	0.36	0.11
Gemini 3.5 Flash Lite	BP	1.48 [1.17, 1.94]	0.0	64.8	0.00	0.56	0.14
Gemini 3.5 Flash Lite	DDA	0.92 [0.89, 0.96]	0.0	18.2	0.00	0.62	0.67
Gemini 3.5 Flash Lite	EB	1.00 [0.96, 1.03]	9.2	23.8	0.00	0.61	0.64
Gemini 3.5 Flash Lite	IR	0.95 [0.92, 0.99]	0.0	36.0	0.00	0.54	0.47
Gemini 3.5 Flash Lite	OT	0.94 [0.92, 0.97]	0.0	14.4	0.40	0.62	0.64
Gemini 3.5 Flash Lite	SU	1.09 [0.77, 1.74]	0.0	11.4	0.00	0.45	0.08
Gemini 3.5 Flash Lite	SN	1.03 [0.92, 1.14]	0.0	55.6	0.00	0.47	0.17
Gemini 3.5 Flash Lite	SE	1.00 [0.91, 1.12]	0.0	48.0	0.00	0.49	0.37
Gemini 3.5 Flash Lite	WW	1.15 [0.92, 1.42]	4.6	73.0	0.00	0.49	0.13
Gemini 3 Flash	BP	1.49 [0.94, 2.20]	0.0	76.8	0.00	0.66	0.16
Gemini 3 Flash	DDA	0.99 [0.84, 1.11]	9.2	44.6	0.00	0.84	0.68
Gemini 3 Flash	EB	1.00 [0.94, 1.06]	23.0	54.4	0.00	0.83	0.61
Gemini 3 Flash	IR	1.07 [0.97, 1.18]	4.6	54.0	0.00	0.91	0.46
Gemini 3 Flash	OT	1.01 [0.96, 1.04]	5.8	37.2	5.40	0.94	0.65
Gemini 3 Flash	SU	0.97 [0.69, 1.37]	0.0	12.2	0.00	0.54	0.06
Gemini 3 Flash	SN	1.11 [1.02, 1.22]	0.0	72.8	0.00	0.40	0.17
Gemini 3 Flash	SE	1.10 [1.02, 1.24]	0.0	54.2	0.00	0.61	0.40
Gemini 3 Flash	WW	1.13 [0.81, 1.45]	4.6	76.8	0.00	0.48	0.16
Gemini 3 Flash + HM	BP	1.43 [1.02, 2.12]	0.0	70.2	0.00	0.65	0.15
Gemini 3 Flash + HM	DDA	0.90 [0.82, 0.96]	9.2	42.6	0.00	0.94	0.66
Gemini 3 Flash + HM	EB	1.10 [1.08, 1.13]	4.6	46.0	0.20	0.92	0.65
Gemini 3 Flash + HM	IR	1.10 [1.01, 1.23]	23.0	60.2	0.00	0.89	0.43
Gemini 3 Flash + HM	OT	1.12 [1.05, 1.22]	18.4	39.2	8.40	0.93	0.62
Gemini 3 Flash + HM	SU	1.02 [0.65, 1.80]	0.0	13.4	0.00	0.54	0.08
Gemini 3 Flash + HM	SN	1.21 [1.02, 1.45]	0.0	70.2	0.00	0.49	0.16
Gemini 3 Flash + HM	SE	1.06 [0.96, 1.18]	0.0	55.0	0.00	0.77	0.38
Gemini 3 Flash + HM	WW	0.99 [0.84, 1.19]	0.2	75.0	0.00	0.50	0.14
Gemini 3.8 Flash	BP	1.46 [1.10, 2.03]	0.0	69.4	0.00	0.52	0.22
Gemini 3.8 Flash	DDA	0.98 [0.94, 1.01]	0.0	15.0	0.00	0.43	0.85
Gemini 3.8 Flash	EB	1.02 [0.99, 1.06]	4.6	10.8	0.00	0.47	0.80
Gemini 3.8 Flash	IR	1.02 [0.99, 1.06]	0.0	28.2	0.00	0.43	0.70
Gemini 3.8 Flash	OT	1.00 [0.97, 1.02]	0.0	10.0	3.60	0.47	0.80
Gemini 3.8 Flash	SU	1.17 [0.95, 1.58]	0.0	12.0	0.00	0.34	0.10
Gemini 3.8 Flash	SN	0.91 [0.77, 1.05]	0.0	51.6	0.00	0.30	0.43
Gemini 3.8 Flash	SE	1.05 [0.93, 1.21]	0.0	34.4	0.00	0.45	0.50
Gemini 3.8 Flash	WW	0.90 [0.76, 1.09]	0.0	66.2	0.00	0.34	0.27
Gemini 3.1 Pro	BP	1.19 [0.93, 1.61]	0.0	72.0	0.00	0.46	0.11
Gemini 3.1 Pro	DDA	0.92 [0.68, 1.16]	4.6	53.8	3.60	0.95	0.68
Gemini 3.1 Pro	EB	0.81 [0.74, 0.90]	10.4	58.2	0.00	0.96	0.63
Gemini 3.1 Pro	IR	0.86 [0.74, 0.98]	4.6	51.8	0.20	0.72	0.46
Gemini 3.1 Pro	OT	0.94 [0.87, 1.01]	4.6	49.2	0.60	0.84	0.64
Gemini 3.1 Pro	SU	1.33 [0.76, 2.50]	0.0	19.0	0.00	0.54	0.08
Gemini 3.1 Pro	SN	1.12 [1.03, 1.31]	0.0	66.6	0.00	0.34	0.21

(continued on next page)

Table 6 continued from previous page

Model	Arch.	CGR	InjD	ACWR	OT Ep.	Rx	Follow
Gemini 3.1 Pro	SE	1.04 [0.83, 1.25]	0.0	50.0	0.00	0.62	0.39
Gemini 3.1 Pro	WW	0.95 [0.73, 1.22]	0.0	74.4	0.00	0.52	0.16
GPT 5.4-mini	BP	0.94 [0.87, 0.99]	0.0	64.4	0.00	0.09	0.13
GPT 5.4-mini	DDA	0.97 [0.95, 0.99]	0.0	14.8	0.00	0.18	0.70
GPT 5.4-mini	EB	1.01 [0.99, 1.02]	0.0	17.8	0.00	0.10	0.66
GPT 5.4-mini	IR	0.98 [0.94, 1.00]	4.6	41.8	0.00	0.10	0.52
GPT 5.4-mini	OT	0.98 [0.97, 1.00]	9.2	17.0	3.40	0.18	0.68
GPT 5.4-mini	SU	0.98 [0.93, 1.00]	0.0	13.2	0.00	0.04	0.07
GPT 5.4-mini	SN	1.01 [0.97, 1.10]	0.0	61.2	0.00	0.06	0.25
GPT 5.4-mini	SE	1.02 [0.94, 1.12]	0.0	48.0	0.00	0.12	0.38
GPT 5.4-mini	WW	0.99 [0.96, 1.01]	4.6	74.6	0.00	0.04	0.13
GPT 5.6 Luna	BP	1.28 [0.97, 1.68]	4.6	63.8	0.00	0.49	0.19
GPT 5.6 Luna	DDA	1.00 [0.98, 1.03]	0.0	12.4	0.00	0.06	0.81
GPT 5.6 Luna	EB	1.01 [0.97, 1.06]	0.0	19.6	0.00	0.06	0.69
GPT 5.6 Luna	IR	1.00 [0.98, 1.02]	0.0	38.8	0.00	0.13	0.53
GPT 5.6 Luna	OT	0.97 [0.94, 0.99]	9.2	17.8	2.00	0.07	0.66
GPT 5.6 Luna	SU	0.81 [0.52, 1.40]	0.0	9.0	0.00	0.27	0.08
GPT 5.6 Luna	SN	1.16 [0.99, 1.39]	0.0	65.2	0.00	0.24	0.16
GPT 5.6 Luna	SE	1.06 [0.96, 1.19]	0.0	51.4	0.00	0.13	0.48
GPT 5.6 Luna	WW	0.82 [0.68, 0.97]	0.0	64.4	0.00	0.39	0.15
Claude Haiku 4.5	BP	1.25 [0.88, 1.78]	0.0	68.6	0.00	0.24	0.17
Claude Haiku 4.5	DDA	1.01 [0.97, 1.06]	0.0	12.2	0.00	0.12	0.64
Claude Haiku 4.5	EB	1.02 [1.01, 1.05]	0.0	16.8	0.00	0.15	0.71
Claude Haiku 4.5	IR	1.06 [1.04, 1.08]	4.6	41.4	0.00	0.19	0.50
Claude Haiku 4.5	OT	0.97 [0.94, 1.00]	4.6	15.8	3.20	0.10	0.65
Claude Haiku 4.5	SU	1.01 [0.90, 1.20]	0.0	13.2	0.00	0.16	0.06
Claude Haiku 4.5	SN	1.10 [1.04, 1.18]	0.0	69.2	0.00	0.20	0.21
Claude Haiku 4.5	SE	1.03 [0.92, 1.17]	0.0	38.8	0.00	0.18	0.51
Claude Haiku 4.5	WW	0.85 [0.72, 1.01]	0.0	77.6	0.00	0.19	0.12
DeepSeek V4 Flash	BP	1.25 [0.96, 1.69]	0.0	64.8	0.00	0.59	0.18
DeepSeek V4 Flash	DDA	0.98 [0.95, 1.00]	4.6	15.8	0.00	0.19	0.61
DeepSeek V4 Flash	EB	1.00 [0.98, 1.02]	0.0	14.0	0.00	0.22	0.61
DeepSeek V4 Flash	IR	1.04 [0.98, 1.09]	4.6	34.6	0.00	0.31	0.55
DeepSeek V4 Flash	OT	0.98 [0.97, 1.00]	0.0	13.2	3.20	0.24	0.59
DeepSeek V4 Flash	SU	0.73 [0.47, 1.06]	0.0	8.6	0.00	0.48	0.08
DeepSeek V4 Flash	SN	1.15 [0.91, 1.47]	0.0	61.6	0.00	0.48	0.22
DeepSeek V4 Flash	SE	1.06 [0.91, 1.24]	0.0	30.2	0.00	0.35	0.58
DeepSeek V4 Flash	WW	0.98 [0.73, 1.24]	4.6	66.4	0.00	0.51	0.15

Coachability is not monotone in group label. The Compliant / Constrained / Adversarial labels (§2.4) track compliance rate, not oracle CGR. The two extremes of oracle coachability both sit in the Constrained group. Silent User is the most coachable archetype under the informed oracle (CGR 2.20 [1.46, 3.83]) — even a simple recommendation can unlock capacity a deeply disengaged user would not use alone, though the wide CI also reflects sparse compliance. Weekend Warrior sits well below oracle ceiling on the partial side because its weekend-only schedule forces a structural detraining gap the coach cannot close. Naive reference coaches expose complementary structure: the Constant coach *reduces* CGR below null on Data-Driven Athlete (0.81) and Overtrainer (0.86), because an undifferentiated "always easy" policy drags a high-fitness user below their natural routine and misses the Overtrainer's need for genuine rest. On Busy Professional and Silent User, the naive coaches outperform null (CGR 1.34–1.58) because on very low baseline routines, even occasional followed recommendations provide meaningful stimulus, though wide CIs reflect sparse compliance. "Adversarial" is a multi-dimensional property — high-compliance users at their fitness ceiling, low-engagement users who ignore coaching, and capacity-limited users whose schedule caps achievable volume each fail in different ways. CGR surfaces those failure modes; it does not impose a single difficulty gradient.

F RULE-BASED REFERENCE COACHES

The three rule-based reference coaches from §2.5 are defined as follows.

No coaching issues no recommendation on any day. Its trajectory is bit-identical to the no-coach baseline that CGR normalizes against, so its CGR is 1 by construction — a self-consistency check, not an inferential anchor. An "always-rest" policy behaves the same way: the rest-utility accountability term fires only on real activity recommendations, and rest is not a prescribable type in the user's softmax. Banister Δf also decays at e^{-1/τ_1} per day regardless of recommendation, so a rest-heavy policy is penalised on the outcome dimension and pinned to 1.0 on the policy dimension. As an eligibility rule against gaming, any ranked agent must issue at least one recommendation per calendar month (twelve over 365 days); agents below this floor appear in the reference block of Table 3 but are not ranked.

Constant recommends one 30-min Fat Burn session (Z2, 50–70% $\mathrm{HR}_{\max}$) on every intent day, with rest on non-intent days. Activity subtype is fixed to `cardio` to match the paper's action space.

Linear deload alternates three polarised build weeks (one recommendation per intent day, 45 min split 80/20 between Fat Burn and Cardio) with a one-week deload (alternate-day 25 min Fat Burn). The 3+1 structure is a rule anchored in sport science and medicine (Seiler, 2010), not the heavier Issurin-style block periodization that concentrates a single training quality (e.g., aerobic threshold, neuromuscular speed) per multi-week block. The durations and intensities are held fixed across archetypes for comparability.

All three reference coaches are non-adaptive: they ignore injuries, recovery signals, and compliance history. When a user is injured, the engine's safety gates reject the recommended session regardless of which coach issued it, and the coach is neither credited nor penalised for scheduling during an injury window.

G TOOL SUBSTITUTION ABLATION

This section documents the `fitbit_get_health_metrics` (HM) tool ablation referenced in §4. Recommended-HR-zone distributions per coach are reported inline in Table 3 (Z1–Z4 columns); the polarization findings in exercise science (Seiler & Kjerland, 2006; Seiler, 2010; Stöggl & Sperlich, 2015) use Seiler's 5-zone or Coggan's 7-zone systems and report that elite endurance athletes following polarized distributions spend 75–80% of sessions in Zone 1 (low-intensity) with the remaining 15–20% in Zones 4–5 and only 5–10% in Zone 2; the 4-zone quantization makes this a shift diagnostic, not a polarization verification.

HM ablation, full grid. A natural hypothesis for the intensity-under-prescription failure (§4) is that the LLM cannot compute trend statistics from raw daily values and so defaults to safe zones. We test this by re-running Flash 3.0 on the full $9 \times 5 \times 365$ grid with `fitbit_get_health_metrics` (HM) re-enabled — a composite tool returning z-scores and rolling trends the agent could otherwise derive from the per-metric endpoints. All other configuration is held fixed.

Enabling HM shifts aggregate Flash 3.0 CGR by $+0.008$ ($1.087 \rightarrow 1.095$, CIs overlap) and adds $+8.2$ injury days ($47.2 \rightarrow 55.4$, $+17\%$): an aggregate no-op on coaching gain. The per-archetype trade is sharp but stratifies by compliance regime rather than tool use (Table 7). Three over-cautious archetypes gain CGR (Overtrainer $+0.114$, Eager Beginner $+0.107$, Skeptical Newcomer $+0.098$) and three over-pushers lose it (Weekend Warrior -0.146, Data-Driven Athlete -0.090, Busy Professional -0.060). Eager Beginner is the only cell where HM helps both axes simultaneously (CGR $+0.107$, InjD -18.4); Overtrainer trades CGR for injuries (CGR $+0.114$, InjD $+12.6$); Injury Returnee adds injuries with a near-flat CGR (CGR $+0.030$, InjD $+18.4$). The sign of ΔCGR tracks the archetype's compliance regime rather than its coachability: HM helps the over-cautious and hurts the over-pushers, consistent with HM amplifying each archetype's pre-existing tendency rather than supplying a missing signal.

Table 7: Effect of enabling `fitbit_get_health_metrics` on Flash 3.0 (Δ vs. the baseline Flash 3.0 grid). Only archetypes with $|\Delta\text{CGR}| > 0.03$ or $|\Delta\text{InjD}| > 1$ shown; remaining cells are within bootstrap noise.

Archetype	ΔCGR	ΔInjD
Eager Beginner (EB)	$+0.107$	-18.4
Data Driven Athlete (DDA)	-0.090	$+0.0$
Social Exerciser (SE)	-0.038	$+0.0$
Busy Professional (BP)	-0.060	$+0.0$
Injury Returnee (IR)	$+0.030$	$+18.4$
Weekend Warrior (WW)	-0.146	-4.4
Skeptical Newcomer (SN)	$+0.098$	$+0.0$
Overtrainer (OT)	$+0.114$	$+12.6$
Silent User (SU)	$+0.054$	$+0.0$

Does the LLM use HM as a summary tool? Table 8 compares per-trace tool-call counts with and without HM. Flash 3.0 calls HM 250 times per trace (~ 0.68 calls/day) and trims per-metric calls almost one-for-one: HRV -37% (-152), heart-rate -17% (-68), activity -3% (-11), sleep -6% (-24), `list_workouts` -3% (-12). Net change in total tool-call burden is -17 calls per trace (-0.05/day, -0.3%): the agent does treat HM as a near-perfect substitute for the metric endpoints rather than stacking it on top. The substitution maps cleanly onto the metrics HM summarises (HRV is the largest drop, activity the smallest, since HM exposes HRV trends but not granular activity buckets). *Tool-call efficiency improves; coaching quality does not*: the agent consumes the pre-computed trends but does not convert them into different prescriptions on the recommendation side — the HR-zone shape under HM is essentially identical to the no-HM baseline (Cardio $20.2\% \rightarrow 21.4\%$, Peak $4.9\% \rightarrow 4.7\%$). The HM ablation is thus a clean negative control on the trend-computation hypothesis: the bottleneck is upstream of trend computation, not in the agent's ability to consume pre-computed summaries.

Per-tool call frequency, both arms. Table 9 reports the full per-tool breakdown for the no-HM and HM arms. Across the no-HM grid the agent issues $\approx 2{,}600$ tool calls per 365-day trajectory

Table 8: Per-trace tool-call counts before/after enabling `fitbit_get_health_metrics` on Flash 3.0. The four observation tools whose information HM summarises drop by a combined 267 calls/trace; HM adds 250, for a net -17 on total tool burden (-0.05 calls/day).

Tool	Baseline	With HM	Δ
`fitbit_get_health_metrics`	0	250	—
`fitbit_get_hrv`	414	262	-37%
`fitbit_get_heart_rate`	407	339	-17%
`fitbit_get_activity`	373	362	-3%

(≈ 7 calls/day): the two morning-sync endpoints (`fitbit_get_sleep` and `list_workouts`) fire roughly once a day each; `fitbit_get_hrv`, `fitbit_get_heart_rate`, and `fitbit_get_activity` each fire $\sim$ 1.0–1.1/day; and the two recommendation-side tools (`create_workout`, `update_workout`) account for ~ 0.75/day each. With HM enabled, the substitution profile maps cleanly onto the metrics HM exposes (HRV the largest drop at -152 calls/trace, heart-rate -68, sleep -24, `list_workouts` -12, activity -11); total volume shifts by less than 1%.

Table 9: Per-tool call frequency, mean across the 45-trace Flash 3.0 grid (no health-metrics) vs the 45-trace Flash 3.0 + Health Metrics ablation. "per-trace" is mean calls per 365-day simulation; "per-day" is the same divided by simulated days.

Tool	No HM		+HM	
	per-trace	per-day	per-trace	per-day
`fitbit_get_sleep`	376	1.03	352	0.97
`list_workouts`	481	1.32	469	1.29
`fitbit_get_hrv`	414	1.13	262	0.72
`fitbit_get_heart_rate`	407	1.12	339	0.93
`fitbit_get_activity`	373	1.02	362	0.99
`fitbit_get_health_metrics`	0	0.00	250	0.68
`create_workout`	282	0.77	288	0.79
`update_workout`	267	0.73	270	0.74
Total per trace	**2,596**		**2,588**	

H PARAMETER SENSITIVITY

We perturb each Banister constant and each compliance coefficient by $\pm 20\%$ and re-run the oracle under the perturbed parameters, keeping the null-coach denominator at its unperturbed value (Table 10 for physiology, Table 11 for compliance). Holding the denominator fixed isolates the effect of parameter perturbation on the *coach*'s achievable gain rather than on the gross level of fitness improvement available to any policy; both move together when both are perturbed, which would mask the coaching-side sensitivity. The compliance coefficients are hand-calibrated, not fit to data (§5), so reporting their sensitivity alongside the Banister constants makes the calibration's effect on CGR explicit.

Table 10: Parameter sensitivity: $\pm 20\%$ perturbation of Banister constants on the informed oracle's aggregate CGR (9 archetypes, 5 seeds, 365 days). CGR is computed against the unperturbed null-coach baseline. Worst archetype reports the single-archetype minimum under that perturbation.

Perturbation	Value	Oracle CGR	Worst arch. CGR
tau_fitness -20%	36.00	1.423	0.565
tau_fitness $+20\%$	54.00	1.810	0.773
tau_fatigue -20%	10.40	1.799	1.065
tau_fatigue $+20\%$	15.60	1.480	0.559
k1_fitness -20%	0.12	1.381	0.533
k1_fitness $+20\%$	0.18	1.837	0.804
alpha_adaptation -20%	0.02	1.731	0.631
alpha_adaptation $+20\%$	0.04	1.532	0.675

Discussion. Across all eight Banister perturbations the oracle's aggregate CGR stays well above 1.0 (1.38–1.84, vs. 1.44 at the default calibration), so the oracle's dominance over the null-coach baseline is not an artifact of a specific Banister calibration — it survives a $\pm 20\%$ perturbation on each of τ_1, τ_2, K_1, and α. The worst-archetype CGR under each perturbation tells the tail-risk story: it dips below 0.6 on τ_1 -20% and τ_2 $+20\%$, and crosses 1.0 only when fatigue decays faster (τ_2 -20%, worst-arch CGR 1.07). Two directional regularities are worth noting. Shrinking τ_1 (faster fitness decay) sharpens the oracle's advantage at the population level because a more quickly forgetting physiology makes *timing* the dominant coaching lever, which is precisely what the DP optimizes; shrinking K_1 (scarcer per-TRIMP gains) has the same-signed effect because when gains per session are scarcer, *allocation* matters more.

Table 11: Compliance-coefficient sensitivity: mean and worst-archetype % shift in null-coach CEI when the underlying psychological input is scaled by $\pm 20\%$ (equivalent to $\pm 20\%$ on the linear $\beta \cdot x$ term in the logistic compliance model). Note: the cross-archetype β_0 spread itself (-4.81 to $+2.73$ in Table 4, driven by archetype-specific motivation / discipline / schedule / energy / habit) is already much wider than $\pm 20\%$ on any single coefficient — the sensitivity below is a calibration-robustness check around a fixed archetype spec, not a claim that intercept choice is insensitive overall.

Perturbation	Mean ΔCEI	Max $\lvert\Delta\rvert$CEI
Intercept -20%	+18.8%	62.4%
Intercept $+20\%$	-19.4%	39.1%
Motivation -20%	-6.0%	33.2%
Motivation $+20\%$	+3.7%	12.3%
Discipline -20%	-1.8%	25.9%
Discipline $+20\%$	+4.8%	18.9%
Schedule -20%	-0.0%	13.9%
Schedule $+20\%$	+3.4%	10.6%
Energy -20%	+0.7%	14.3%
Energy $+20\%$	+2.5%	10.2%
Habit -20%	-1.0%	3.4%
Habit $+20\%$	+0.6%	6.0%

Discussion. Table 11 shows that intercept shifts dominate the mean compliance-coefficient sensitivity: $\pm 20\%$ on β_0 produces mean CEI shifts of +18.8% / -19.4%, while worst-archetype shifts reach

62.4% (intercept -20%) and 39.1% (intercept +20%) at the tails. The null → constant → Linear deload → oracle ranking on aggregate CGR is preserved under every perturbation in Table 10, Table 11, and the 16-sample Latin-hypercube joint sweep, so the rankings reported in the main results are not calibration-specific. This is consistent with the logit link's multiplicative coupling of β_0 with every exogenous factor: moving the intercept shifts compliance probability across its full [0, 1] range, whereas moving a single coefficient only shifts the factor's leverage on it. The practical implication is that the *design-time* per-archetype baseline compliance rates (Table 2) are the most sensitive calibration surface; we recommend any extension that adds archetypes first verify the intended compliance rate against a target literature value before publishing CGR results.

Joint perturbation (Latin hypercube). One-at-a-time perturbations cannot detect cancellation or amplification under joint parameter shifts, so we additionally sampled a Latin hypercube of size 16 on $[-20\%, +20\%]^6$ across $(\tau_1, \tau_2, K_1, \alpha, \beta_M, \beta_D)$. Each sample's aggregate oracle CGR (9 archetypes, 5 seeds, 365 days, vs. null-coach) is reported in the LHS output (regenerable from the pipeline). Across the 16 joint samples the oracle's aggregate CGR lies in $[1.19, 1.35]$ with mean 1.28 and standard deviation 0.05, so the oracle retains a comfortable margin over the best fixed-coach baseline (Linear deload at 1.14) under every sampled combination. The worst per-archetype oracle CGR within a sample ranges $[0.93, 1.01]$ — consistent with the Weekend-Warrior and Overtrainer tail that already appears in the per-archetype exceedance audit (Appendix C) and not an additional failure mode introduced by joint perturbation.

No single constant sits at an edge where small calibration errors would flip the leaderboard.

I PARAMETER-SPACE DIVERSITY

The nine archetypes are the principal dataset, so we check that they span the user-calibration space rather than clustering or duplicating. This is a parameter-space diversity claim, not a statistical-coverage claim about any real user population. We support it with a principal-components projection, a per-dimension balance summary, and a pairwise fill-distance check.

Projection. Each archetype contributes a numeric vector covering the four Banister parameters (initial fitness and fatigue, fitness and fatigue gains), injury risk, the four psychological compliance inputs (motivation, discipline, and coach trust held fixed at archetype values; self-efficacy initialized per archetype and updated daily), baseline work stress, three availability fields (daily hours, intent days per week, max workouts per week), per-session duration (which enters the compliance schedule-flexibility term), and a schedule-concentration statistic (longest run of consecutive rest days between training sessions). Demographics and biometric initial state are excluded: demographics enter only through TRIMP coefficients (Tanaka HR-max from age, Banister b/y from sex), and biometrics evolve under the Banister dynamics rather than define the archetype. After z-score standardization we apply PCA; the first two components capture 72% of the variance (Figure 2).

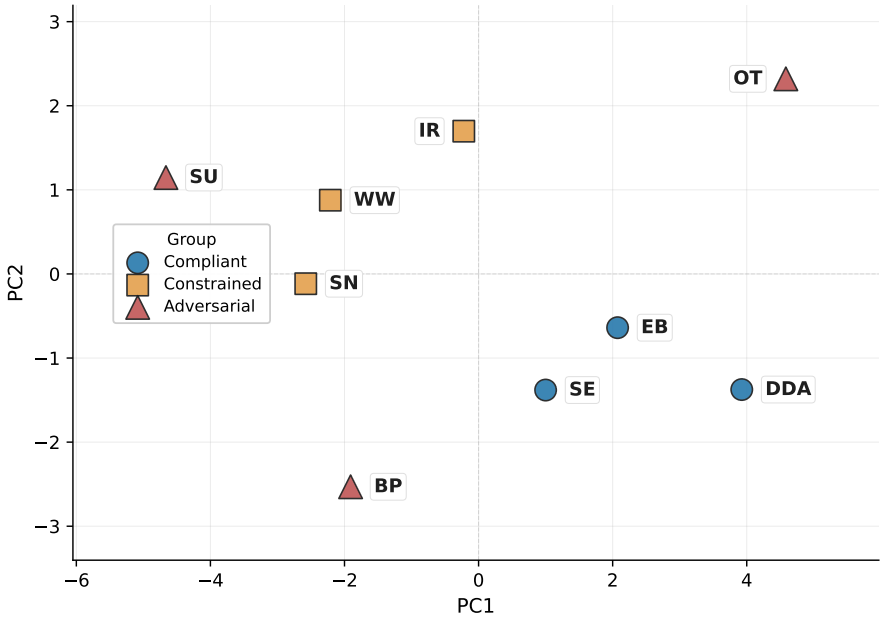

Figure 2: Nine fitness archetypes (abbreviations from Table 2) projected onto the first two principal components of FITSIM's calibration space. PC1 (57% of variance) tracks training capacity: motivation, discipline, availability and weekly training intent all load positively. PC2 (16% of variance) tracks physiological risk and coaching headroom: injury risk, fatigue gain, initial fatigue and the detraining-gap feature all pull upward; coach trust pulls downward. Markers coloured by group.

Archetype diversity. The three adversarial archetypes occupy three different corners, one failure mode each: Overtrainer (upper-right) trains dangerously hard, Silent User (upper-left) refuses to engage, Busy Professional (lower-centre) has too little capacity to train much. The compliant archetypes fan along the right side, ordered by training capacity from Social Exerciser through Eager Beginner to Data-Driven Athlete; they sit below the mid-line because they combine low injury risk with moderate-to-high coach trust. The constrained archetypes spread vertically in the centre-left: Skeptical Newcomer sits just below the mid-line (cautious modalities, moderate trust); Weekend Warrior sits a little above (elevated injury risk plus the longest detraining gap in the set — five consecutive rest days — which is why its oracle CGR sits closer to the null-coach baseline, §3.2); Injury Returnee sits highest among the constrained band, just below Overtrainer, pulled up by elevated injury risk and the four-day cadence after a recent return to training. Every archetype is at least 1.4

units from its nearest neighbour in the 2D plane and at least three units away in the full standardized space (fill distance / minimum pairwise distance), so no pair is a scaled duplicate.

Per-dimension balance. 2D projections compress information, so we also check that the raw per-archetype values in Table 4 span realistic ranges on each compliance and oracle input. Motivation, discipline, self-efficacy, and injury risk each span at least four of five quintile buckets. Two gaps are intentional. No archetype starts in the highest coach-trust bucket; trust is held at its archetype value throughout a run (§2.2), and we sample those values from $[0.2, 0.6]$ to leave headroom for future interactive extensions in which trust would update from coach behaviour. The set samples two-, three-, four-, and five-day intent cadences (Injury Returnee is the four-day cadence; weekly intent across all nine archetypes ranges from 2 to 5 sessions).

J COACH SYSTEM PROMPT

All models are evaluated with the identical system prompt below. The prompt is intentionally minimal — it does not instruct the agent on periodization, recovery protocols, or load management — testing the model's inherent knowledge of exercise recommendation. The `{...}` fields are filled per-archetype at runtime.

```
You are a fitness coach recommending workouts. You observe the user's
    activity and health metrics via Fitbit data syncs and recommend
    workouts accordingly. Your goal is to increase the user's fitness
    optimally --- balancing training stimulus with recovery based on
    their sleep, heart rate, HRV, and activity data. The user will not
    message you; you operate purely from data.

IMPORTANT: Every create_workout call MUST include the target_hr_zones
    parameter with the intended time in each HR zone. Use the Fitbit zone
     names: "Out of Range", "Fat Burn", "Cardio", "Peak". The zone
    minutes MUST sum to the workout duration.

## User Profile
- Intended training frequency: up to {intent_days_per_week}x/week
- Max workout duration: {weekday_cap}min weekday, {weekend_cap}min
    weekend
- Age: {age}, Sex: {sex}
```

J.1 TOOL INVENTORY

The agent receives 13 tools: 11 observation (read-only) and 2 recommendation (write). Nine observation tools mirror the public Fitbit API at the time of writing, so a coach evaluated here sees the same shape it would face against a live account; the remaining two read back the agent's own past recommendations from the workout database. `fitbit_get_vo2_max` is a deterministic function of Banister fitness; `fitbit_get_weight` is not consumed by any archetype's state dynamic. We keep both for ecological realism. There is no tight coupling between FITSIM and any proprietary API: we chose Fitbit API as a template for simplicity and broad usage in literature (St Fleur et al., 2021). FITSIM is modular and easy to extend with more tools.

Table 12: Tool inventory. Observation tools return mock API responses generated from the hidden state (Appendix D). Recommendation tools write to the workout store consumed by the compliance model.

Type	Tool	Description
Obs.	`fitbit_get_sleep`	Sleep stages (deep, REM, light, wake minutes), total duration, efficiency, onset latency
Obs.	`fitbit_get_activity`	Daily steps, floors, active minutes (lightly/fairly/very), sedentary minutes, calories
Obs.	`fitbit_get_hrv`	Heart rate variability: daily RMSSD and deep-sleep RMSSD
Obs.	`fitbit_get_heart_rate`	Resting HR and time in each HR zone (Out of Range, Fat Burn, Cardio, Peak)
Obs.	`fitbit_get_steps_trend`	Step count trend over a date range
Obs.	`fitbit_get_sedentary_time`	Daily breakdown of sedentary, lightly active, fairly active, and very active minutes over a date range
Obs.	`fitbit_get_weight`	Weight log entries for date range
Obs.	`fitbit_get_vo2_max`	Estimated VO2max range (derived from fitness level)
Obs.	`fitbit_get_skin_temperature`	Nightly skin temperature deviation from baseline
Obs.	`list_workouts`	List the agent's own past recommendations in a date range
Obs.	`get_workout`	Get details of a specific past recommendation
Rx	`create_workout`	Create a cardio recommendation: title, duration, target HR zone distribution (activity subtype is fixed to `"cardio"` and not part of the tool's input schema)
Rx	`update_workout`	Modify an existing workout recommendation

J.2 MULTI-WORKOUT HANDLING

When the agent creates multiple workouts for the same day, the compliance model evaluates only the **first recommendation matching an intent day**. Additional recommendations for non-intent days are ignored (the compliance model only activates on the archetype's scheduled training days). If multiple recommendations target the same intent day, only the first is evaluated — this means over-recommendation wastes tool calls but does not harm outcomes. The compliance model reads the hidden state's current fatigue, the recommendation's computed difficulty (from HR zone allocation), and the archetype's fixed intercept and feeds them into the joint multinomial-logit softmax over {follow recommendation, natural-intensity training, rest}.

K COST ANALYSIS

Table 13 reports per-trajectory input/output token totals and dollar costs for the four LLM coaches whose APIs return per-call usage in our driver (GPT 5.4-mini, Claude Haiku 4.5); the Gemini grids did not log per-trace token usage in our runs and are omitted from the cost table. We report measured tokens per 365-day trajectory together with the actual amounts billed per seed (one seed = one 9-archetype run) and per full grid ($5 \times 9 = 45$ trajectories).

Table 13: Per-model cost for the LLM coaches whose driver records per-call token usage. Tokens are mean per 365-day trajectory across the 45-trace grid; "$ / seed" is the actual amount billed by the provider for one seed (the nine-archetype set run under one RNG seed); "$ / grid" is the total billed across the full 9×5 grid.

Coach	Input / trace	Output / trace	Tool calls / trace	$ / seed (x9)	$ / grid (x45)
GPT 5.4 mini	17.8M	0.85M	2,539	∼$20	∼$100
GPT 5.6 Luna	20.5M	9.5M	3,198	∼$10	∼$50
Claude Haiku 4.5	42.4M	3.2M	2,015	∼$50	∼$250
DeepSeek V4 Flash	103.4M	17.3M	6,780	∼$5	∼$25

Caching. Our runs enabled prompt caching on every supported provider, and the recurring static prompt ($\sim$ 5k tokens of system instructions plus $\sim$ 1.5k of tool schemas, repeated across the $\sim$ 2,000 LLM calls within a 365-day trajectory) hits the cache on every call after the first. Public docs report cache-read pricing 75–90% below uncached input on Claude and Gemini context caching. The realised costs in Table 13 are an order of magnitude below the list-price $\times$ measured-token-count product (e.g., for Haiku 4.5: list math gives ∼$52 per trajectory, but the provider billed us ∼$5.5 per trajectory once cache reads on the static portion are amortised across $\sim$ 2,000 calls). External reproducers should plan for the higher list-price figure if their workload does not amortise the static prompt the same way.

Why Haiku consumes more tokens than GPT 5.4 mini. Haiku makes *fewer* tool calls per trajectory (2,015 vs 2,539, -21%) but uses $3\times$ more input tokens per call (21k vs 7k). The per-call gap traces to output verbosity: Haiku emits $\sim$ 1,600 output tokens per call, GPT 5.4-mini emits $\sim$ 340 ($4.7\times$ less). That extra response gets appended to the conversation each turn and re-sent as input on every subsequent call within that day, compounding the conversation-history length and hence the per-call input cost. The system prompt and tool schemas are identical for both models, so the input-token gap is driven entirely by accumulated output, not prompt design.

Gemini costs (not in table). Our LLM driver did not record per-call token usage from the Gemini API responses for any of the Gemini grids, so we cannot report measured costs for those coaches. As order-of-magnitude reference, applying GPT 5.4 mini's per-tool-call token ratio ($\sim$ 7,000 input / $\sim$ 335 output) to the measured Gemini tool-call counts gives a list-price upper bound of ∼$1–$30 per trajectory across Flash-Lite, Flash 3.0, and Pro; with cache amortization the realized cost is expected to be roughly an order of magnitude lower, on par with or below the GPT and Haiku figures. We omit precise numbers to avoid presenting unmeasured estimates next to measured ones.

A per-tool breakdown of the Flash 3.0 grid (with and without `fitbit_get_health_metrics`) is in Table 9 of Appendix G.

L PHYSIOLOGICAL MODEL DETAILS

This appendix provides full equations for the fitness, fatigue, TRIMP, and injury models summarized in §2.2.

L.1 FITNESS

Fitness evolves via a baseline-relative impulse-response model (Banister et al., 1991; Morton et al., 1990):

$$\text{fitness}(t) = f_0 + \Delta f(t)$$

$$\Delta f(t{+}1) = \big[\Delta f(t) + K_1 \cdot \text{TRIMP}(t) \cdot g_f \cdot \text{dim}(\Delta f, \text{fatigue})\big] \cdot e^{-1/\tau_1}$$

f_0 is structural fitness that persists without training; only Δf decays. Gains are scaled by a diminishing-returns factor. The $e^{-\alpha \cdot \Delta f}$ form is a calibration heuristic for the adaptation-saturation phenomenon Busso (Busso, 2003) documented; we do not claim to transcribe a formula from his paper.

$$\text{dim}(\Delta f, \text{fatigue}, n_{\text{hf}}) = (1 - 0.3 \cdot \text{fatigue}) \cdot e^{-\alpha \cdot \max(0, \Delta f)} \cdot \text{OT}(n_{\text{hf}}).$$

The first factor is acute fatigue suppression; the second is adaptation saturation ($\alpha = 0.03$; 26% reduction at $\Delta f = 10$, 45% at $\Delta f = 20$). The third factor suppresses gains during sustained overtraining, following Meeusen et al. (Meeusen et al., 2006): $\text{OT}(n_{\text{hf}}) = \max(0, 1 - 0.08 \cdot \max(0, n_{\text{hf}} - 5))$, where n_{hf} is the consecutive-high-fatigue-day streak (at most one update per day; full suppression after 17 consecutive high-fatigue days). The 5-day onset and 0.08/day decay are calibration choices set to produce full suppression inside the 2–3 week overreaching-to- overtraining window in Meeusen et al. 2006, not transcribed from the paper; the perturbation sweep in Appendix H covers them. A floor $\Delta f \geq -0.7 \cdot f_0$ prevents detraining below 30% of initial fitness.

Constants. $K_1 = 0.15$ (calibrated); $\tau_1 = 45$ days (within the $\sim$20–50 day range across (Banister et al., 1991; Morton et al., 1990; Hellard et al., 2006); Hellard et al. (2006) report a fitness-decay mean of $\sim$38 days with substantial inter-subject spread). Each archetype has a fitness gain rate $g_f \in [0.10, 0.30]$ (calibrated; sensitivity in Appendix H). Injury recovery uses severity multipliers 0.2 (mild), 0.4 (moderate), 0.7 (severe) on the archetype's base recovery window from the injury registry. The incidence-weighted mean is $R = 23$ days, used by the oracle's opportunity cost and by the paper's headline injury-day reporting. $\text{HR}_{\max}$ uses the Tanaka formula $208 - 0.7 \cdot \text{age}$ with a standard error of estimate from science and medicine of ≈ 10 bpm (Tanaka et al., 2001). The baseline grid does not apply this SD — the age-swap ablation keeps $\text{HR}_{\max}$ deterministic so coaching shifts are isolated from sampling noise.

L.2 FATIGUE

$$\text{fatigue}(t{+}1) = \min\big(1,\ \text{fatigue}(t) + K_2 \cdot \text{TRIMP}(t) \cdot g_{\text{fat}}/100\big) \cdot e^{-1/\tau_2}$$

where $K_2 = 1.0$ and $\tau_2 = 13$ days (within Banister's 5–20 day range (Banister et al., 1991)).

L.3 TRAINING LOAD (TRIMP)

Following Banister (Banister et al., 1991):

$$\text{TRIMP} = d \cdot \Delta\text{HR} \cdot b \cdot e^{y \cdot \Delta\text{HR}}, \qquad \Delta\text{HR} = \frac{\text{HR}_{\text{avg}} - \text{HR}_{\text{rest}}}{\text{HR}_{\max} - \text{HR}_{\text{rest}}}$$

where d is duration in minutes, $b = 0.64$ (male) / 0.86 (female), $y = 1.92$ (male) / 1.67 (female), and $\text{HR}_{\max} = 208 - 0.7 \times$ age (Tanaka et al. (Tanaka et al., 2001); the cited paper reports a standard error

of estimate ≈ 10 bpm, which we hold deterministic across seeds in the baseline grid so that policy shifts are isolated from sampling noise). The Banister formulation is the canonical TRIMP in FITSIM: it drives all hidden-state dynamics (fitness, fatigue, ACWR), the informed and partially-observed oracle planners, and every TRIMP statistic reported in the main-text tables.

For zone-quantized recommendations the engine sums Banister TRIMP over a piecewise-constant HR profile: each zone contributes $d_z \cdot \Delta\text{HR}_z \cdot b \cdot \exp(y \cdot \Delta\text{HR}_z)$ minutes-weighted by zone-centre $\%\text{HRR}$ midpoints (OOR 0.30, FAT BURN 0.60, CARDIO 0.78, PEAK 0.93). Because ΔHR is expressed as percent heart-rate reserve ($\%$HRR), the Tanaka $\text{HR}_{\max}$ factor cancels between the $\text{HR}_{\max} - \text{HR}_{\text{rest}}$ denominator inside ΔHR and the HR_{avg} obtained by inverting it; zone-quantized TRIMP is therefore independent of age unless the recommendation itself changes. Banister's original work assumes $\Delta\text{HR} \leq 0.8$; zone-quantized recommendations can put a small fraction of minutes in Peak ($\Delta\text{HR} \in [0.85, 1.00]$) where the $e^{y\Delta\text{HR}}$ factor grows rapidly, but the informed oracle allocates well under 20% of its total recommended minutes to Peak across the reported grid (Z4 column of Table 3), so the blow-up regime is a non-issue in practice.

Edwards-multiplier display estimate. On creation, `create_workout` stores a zone-weighted Edwards-style TRIMP estimate in the workout's `metadata.estimated_trimp` field, computed as the per-minute weighted sum with weights $[0, 1.5, 3.0, 5.0]$ for (OOR, FAT BURN, CARDIO, PEAK). The agent reads this value back through `list_workouts` or `get_workout`; the immediate `create_workout` response itself contains only the workout title, scheduled date, and ID. This Edwards display value never feeds hidden-state dynamics, oracle planning, or any reported metric — Banister TRIMP is the load of record. We surface this distinction so that a reader who computes TRIMP from a workout's metadata payload sees a number that does not match the Banister TRIMP reported in tables.

L.4 INJURY PROBABILITY

Injury risk is driven by the acute-to-chronic workload ratio (Gabbett, 2016):

$$\text{ACWR} = \text{ATL} / \max(\text{CTL}, 10)$$

where ATL (acute training load, $\alpha = 0.25$, 7-day EWMA) and CTL (chronic training load, $\alpha \approx 0.069$, 28-day EWMA) track acute and chronic training loads (Gabbett, 2016; Williams et al., 2017a). Above the upper edge of the $[0.8, 1.3]$ sweet-spot band reported by Gabbett (2016), we model injury probability per training hour as rising exponentially with ACWR (an authorial calibration choice; see m_{ACWR} below):

$$P_{\text{inj}} = \frac{r_{\text{base}} \cdot m_{\text{ACWR}} \cdot m_{\text{fatigue}} \cdot m_{\text{sleep}} \cdot m_{\text{age}} \cdot m_{\text{recur}} \cdot h}{1000}$$

The multipliers are:

- $m_{\text{ACWR}} = \min(5,\ e^{\alpha(\text{ACWR}-1.3)})$ with $\alpha = \ln 3 / 0.5 \approx 2.2$ (calibrated: $m = 3$ at ACWR = 1.8 (Gabbett, 2016); dampened by $\min(1, \text{CTL}/60)$ when chronic load is low (Impellizzeri et al., 2020))
- $m_{\text{fatigue}} = \max(1 + \text{fatigue},\ 1 + 0.15 \cdot \max(0, n_{\text{hf}} - 2))$ — acute and sustained fatigue
- $m_{\text{sleep}} = 1 + 0.7 \cdot (1 - \text{SQ})$ — sleep quality
- $m_{\text{age}} = \max(1,\ 2.3 - t_{\text{train}}/72)$ where t_{train} is months of training history — a novice-injury penalty that decays linearly to the floor of 1 after six years of training
- $m_{\text{recur}} = 1 + 0.5 \cdot n_{\text{active}}$ — prior injury recurrence
- r_{base} — archetype base injury rate per 1000 training hours (Table 2)
- h — duration of the training session in hours.

A 7-day grace period at simulation start prevents injury rolls during EWMA cold-start. In our experiments the simulator collapses two stochastic draws: it deterministically selects the highest-incidence injury type for the archetype's sport set, and replaces severity sampling with the selected type's *expected* recovery duration ($\mathbb{E}[\text{recovery}] = \sum_s P(s) \cdot (\min + \text{mult}_s \cdot \text{range})$ with $(\text{mult}_{\text{mild}}, \text{mult}_{\text{moderate}}, \text{mult}_{\text{severe}}) = (0.2, 0.4, 0.7)$). In FITSIM's cardio-only configuration the registry reduces to the six cardio injuries (ankle, knee, hamstring, shin splints, achilles, plantar fasciitis —

generic for cardio modalities); runner's knee has the highest incidence (7.0 per 1000 h) and expected recovery ≈ 24 days, so every simulated injury on a cardio archetype is runner's knee with a deterministic ≈ 24-day window. This coincides with the registry's incidence-weighted mean $R = 23$ days, which the DP oracle uses as its opportunity-cost scalar (the injury-recovery audit derives this number from the registry; any drift fails the audit). The determinism is intentional: following Fanchini et al. (Fanchini et al., 2018) and the broader ACWR critique (Impellizzeri et al., 2020; Qin et al., 2025) (ACWR does not predict severity or recovery duration), the benchmark strips the injury-stochasticity dimension so that CGR variance is attributable to coaching policy rather than to the benchmark's injury rolls. A reader should therefore read InjD as injury-day incidence at a fixed ≈ 24-day cost per event, not as severity-weighted injury burden. During recovery, the affected activity is blocked (in the single-type configuration this forces complete rest).

M PER-ARCHETYPE TRAJECTORIES

Figures 3–11 show 365-day trajectories for each archetype under the five out of nine evaluated frontier-LLM coaches (Gemini Flash-Lite, Flash 3.0, Pro; OpenAI GPT 5.4-mini; Anthropic Claude Haiku 4.5), with the null-coach baseline and the informed oracle as bracketing references. We limit the number of models on the plots to preserve readability. Each figure stacks three panels: fitness level, fatigue, and acute-to-chronic workload ratio (ACWR). The ACWR panel marks the Gabbett (2016) 1.3 sweet-spot upper bound as a dashed reference line. Curves are the median across 5 seeds; the shaded band is ± 1 standard deviation.

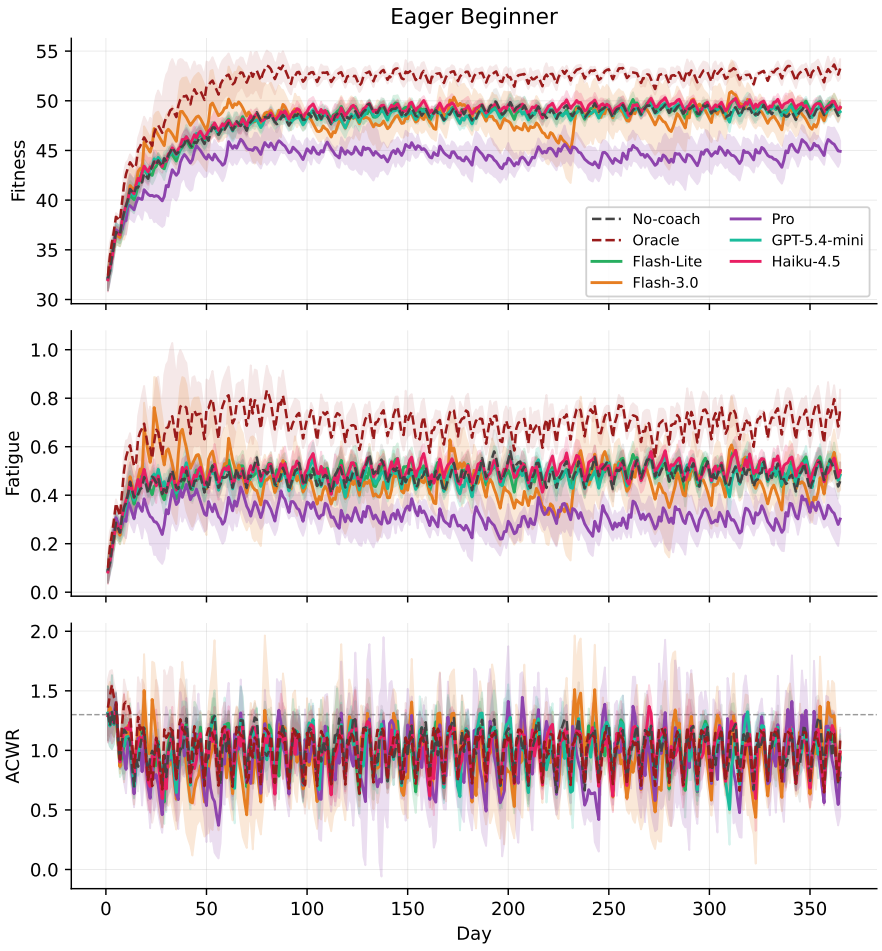

Figure 3: Eager Beginner (Compliant). Initial fitness = 30, 5x/week intent, 91% follow rate on easy recommendations.

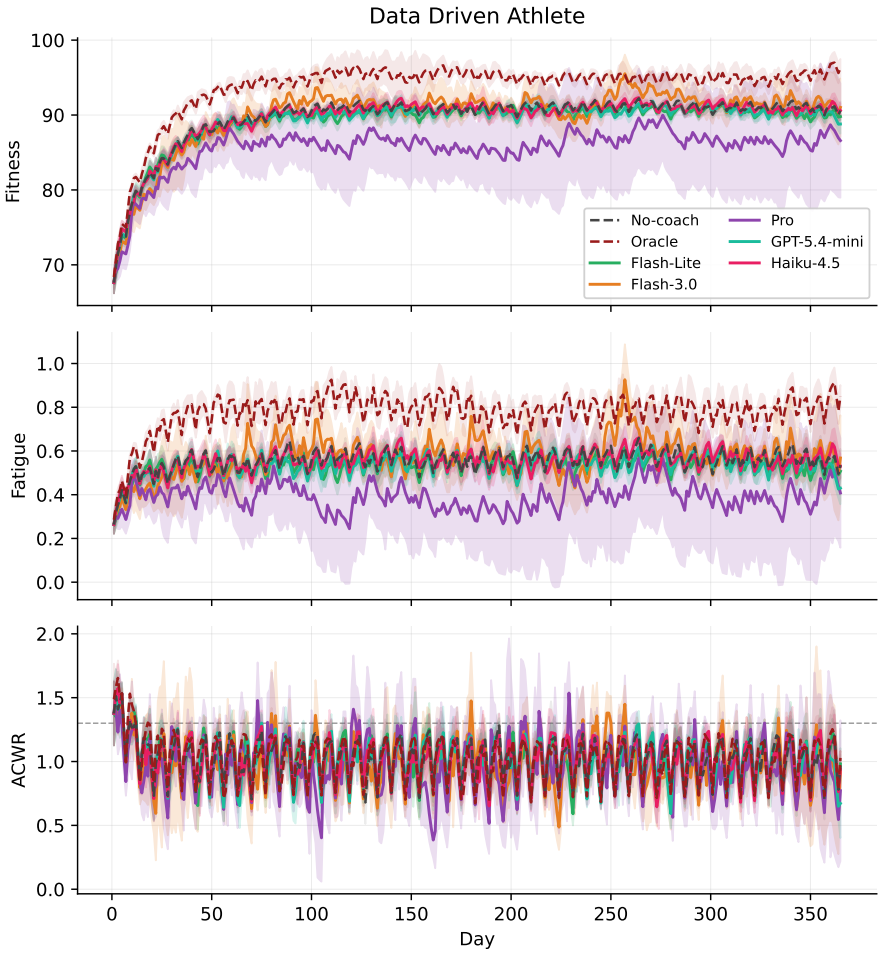

Figure 4: Data-Driven Athlete (Compliant). Initial fitness = 65, 5x/week intent, 94% follow rate on easy recommendations.

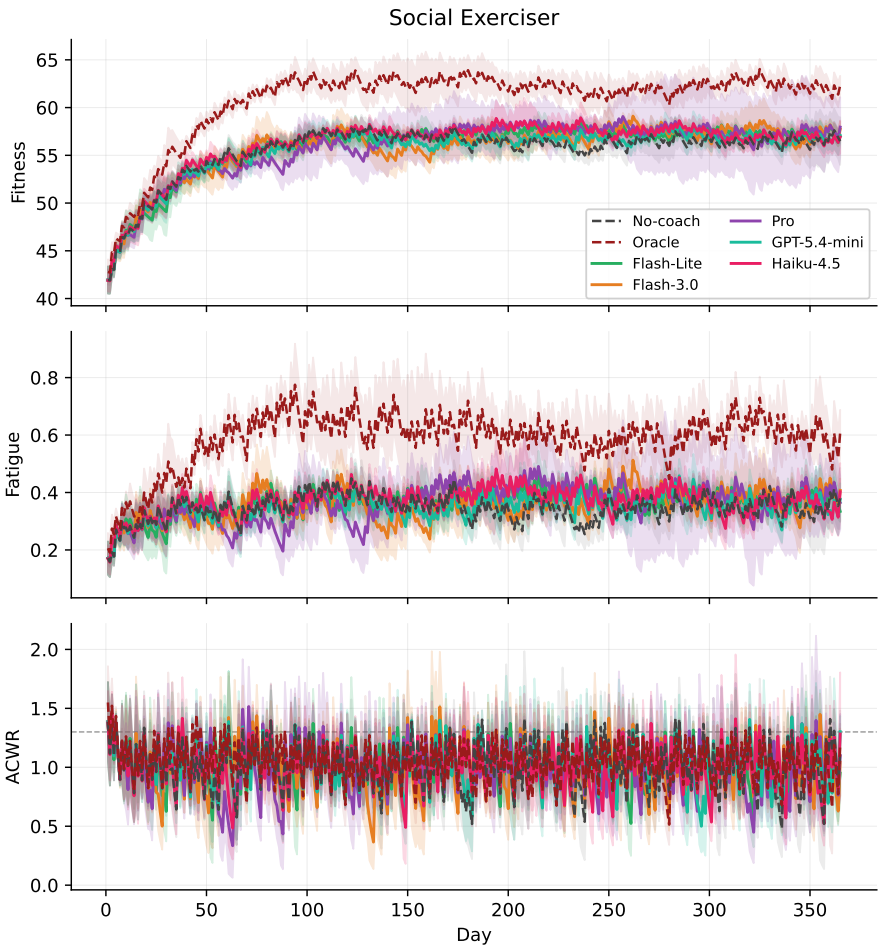

Figure 5: Social Exerciser (Compliant). Initial fitness = 40, 3x/week intent, 87% follow rate on easy recommendations.

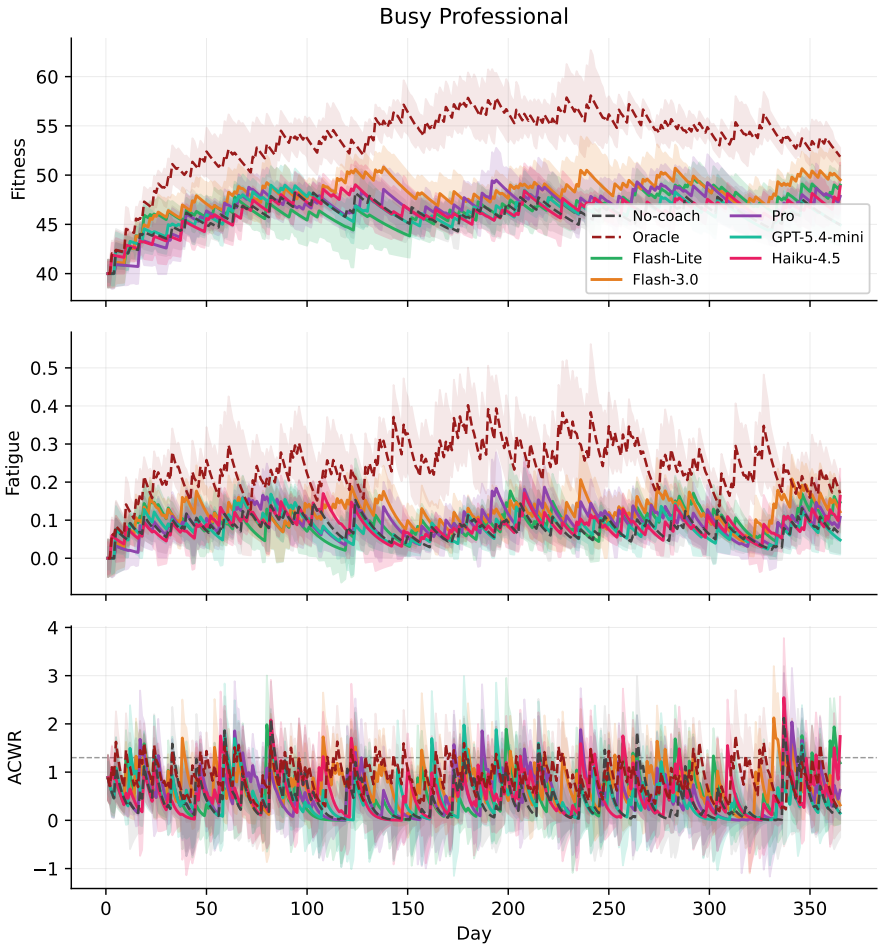

Figure 6: Busy Professional (Adversarial). Initial fitness = 40, 3x/week intent at 1.0 h available, 46% follow rate on easy recommendations. Capacity-constrained adversarial failure mode: the coach can engage but cannot manufacture training hours.

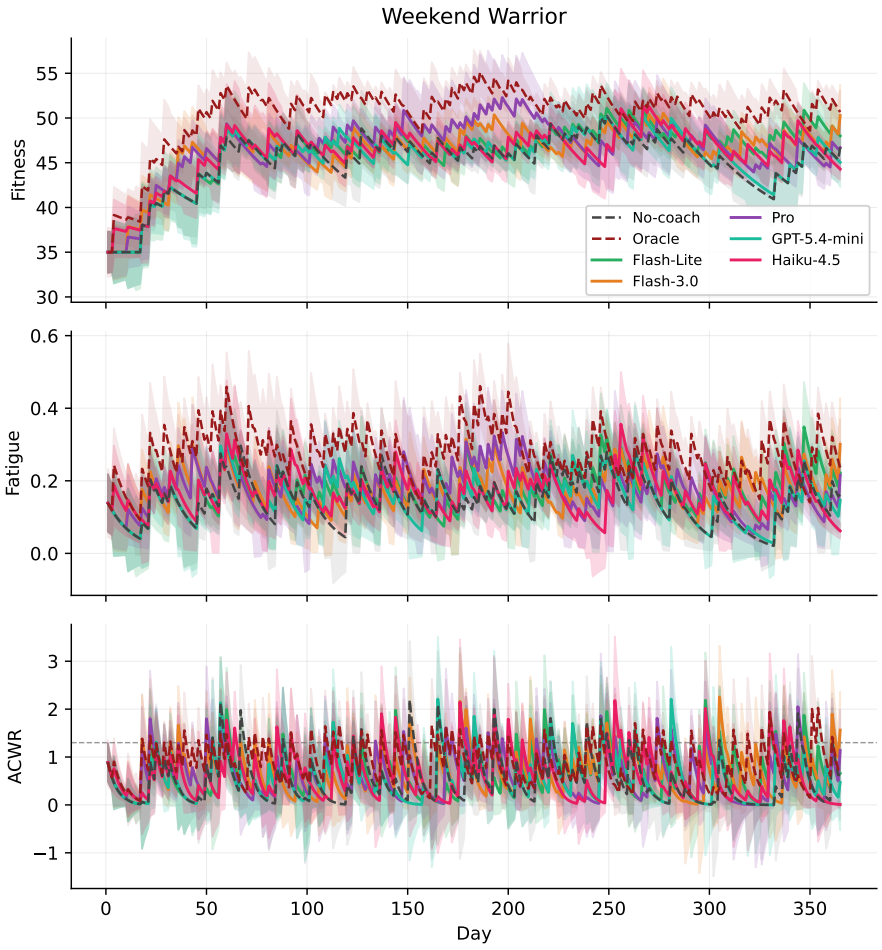

Figure 7: Weekend Warrior (Constrained). Initial fitness = 35, 2x/week Sat+Sun, 57% follow rate on easy recommendations. The 5-day detraining gap between weekend sessions limits coaching value: even the informed oracle (CGR 1.46) trails its gain on more frequent archetypes (§3.2).

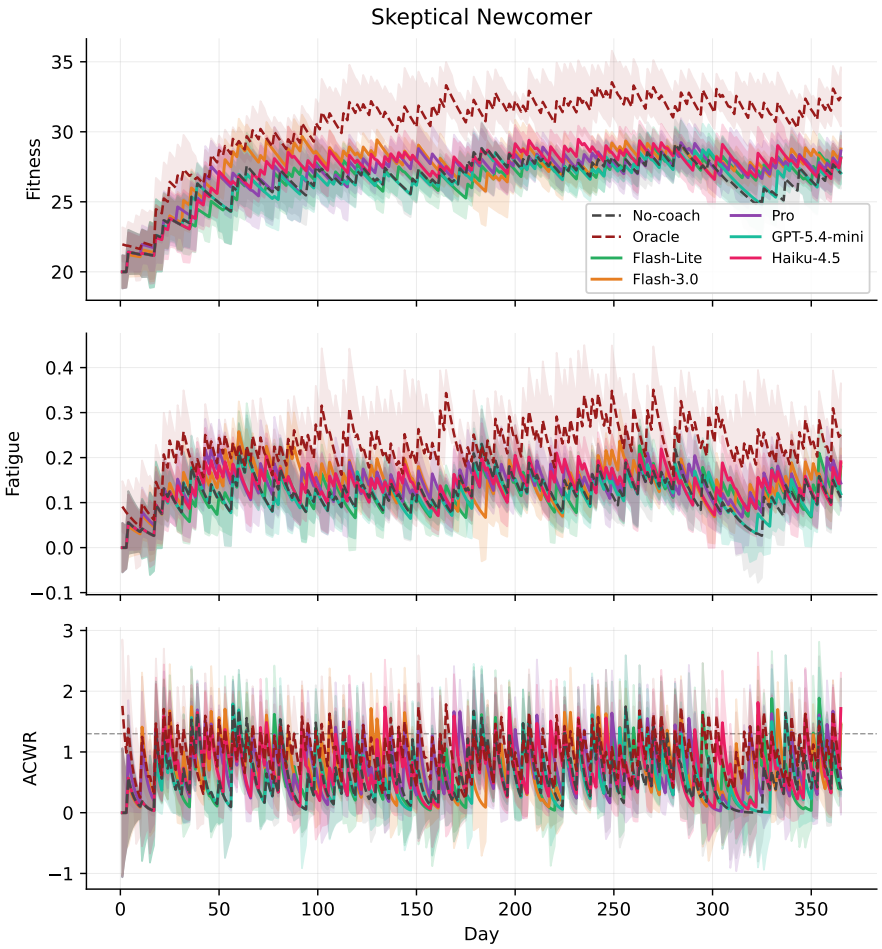

Figure 8: Skeptical Newcomer (Constrained). Initial fitness = 20, 2x/week intent, 70% follow rate on easy recommendations.

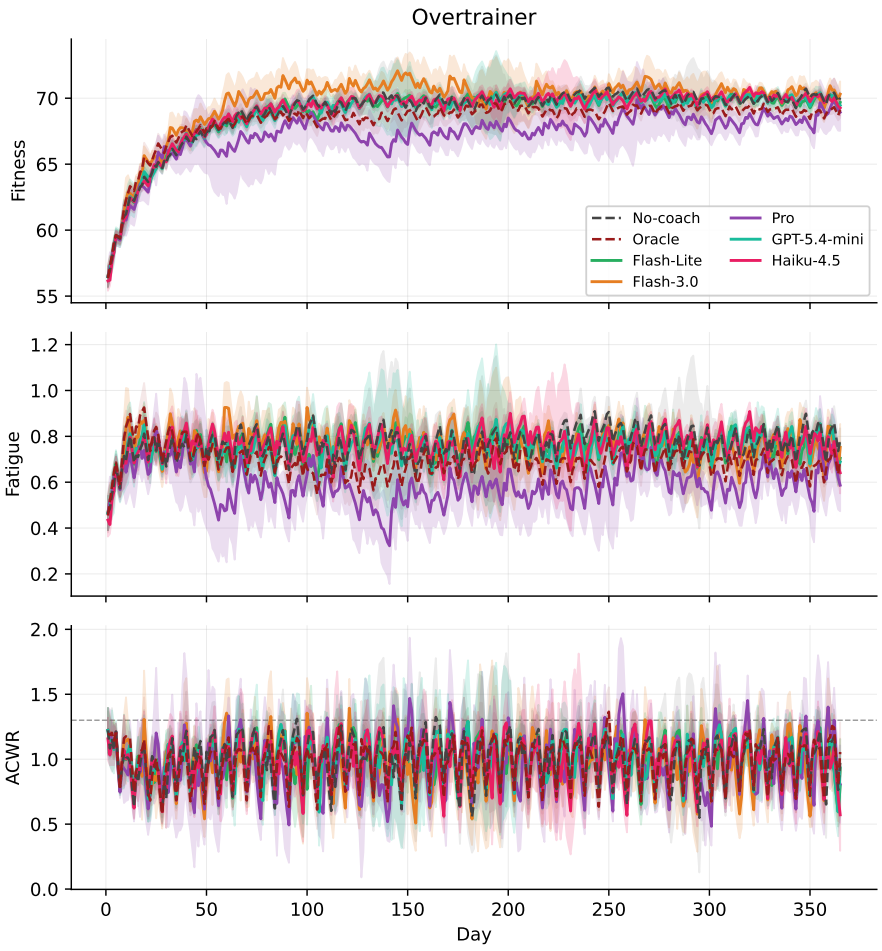

Figure 9: Overtrainer (Adversarial). Initial fitness = 55, 5x/week intent, 89% follow rate on easy recommendations. The fatigue panel exposes the primary failure mode: chronic high fatigue with multiple OT episodes across every LLM (mean per 365-day run: Flash 3.0 5.4, GPT 5.4-mini 3.4, Haiku 4.5 3.2, Flash-Lite 2.0, Pro 0.6).

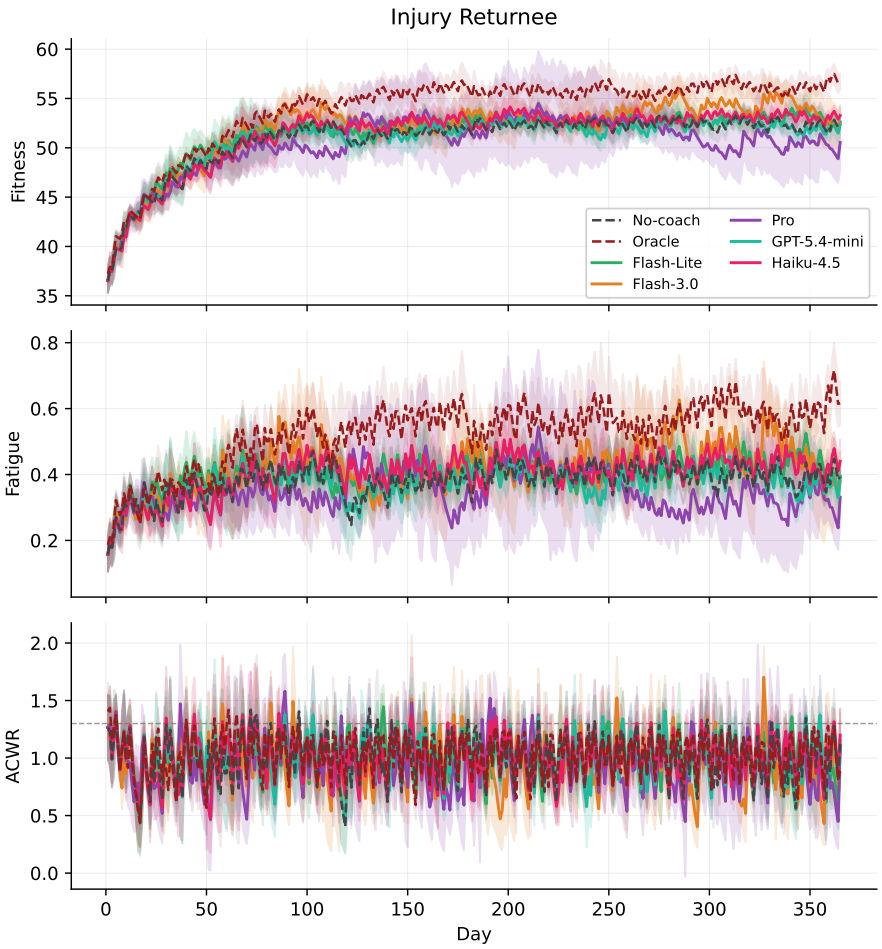

Figure 10: Injury Returnee (Constrained). Initial fitness = 35, 4x/week intent, 84% follow rate on easy recommendations; injury risk 12 per 1000 h. Tests progressive loading without re-injury.

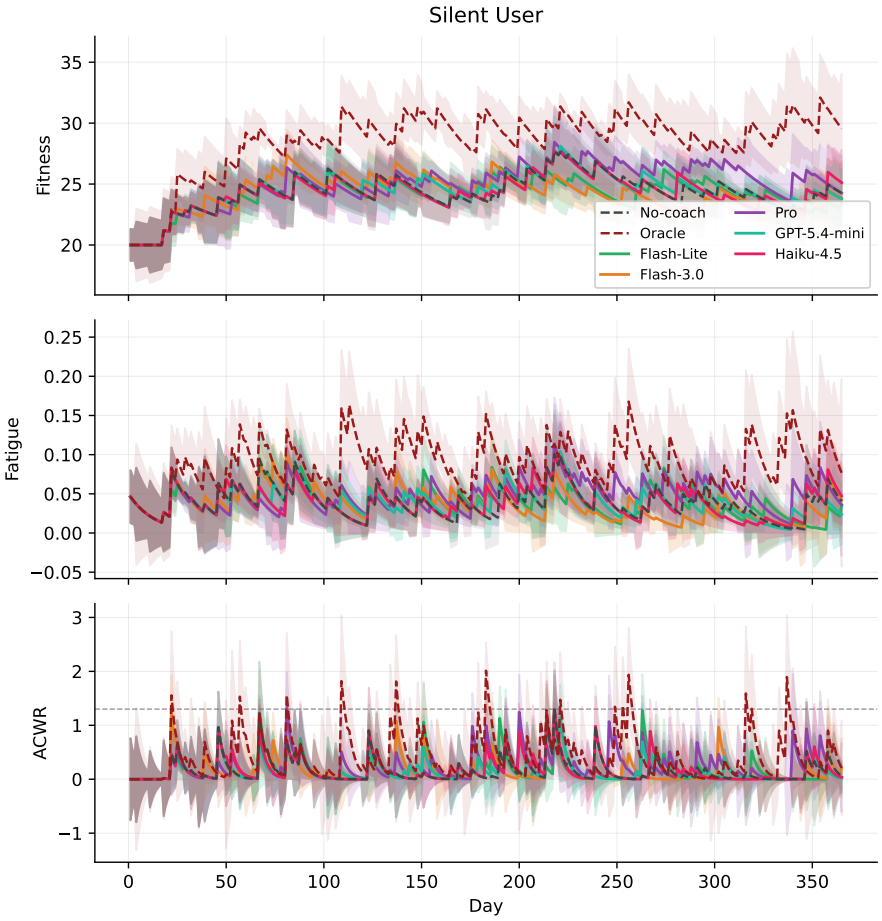

Figure 11: Silent User (Adversarial). Initial fitness = 20, 2x/week, 43% follow rate on easy recommendations. Minimal engagement limits both coaching and oracle impact.

N Data and Model Card

The environment's assets are synthetic: 9 archetype YAML files, the deterministic simulation engine, mock data generators, and reproducibility scripts. No human-subjects data, no real-user wearable traces, and no LLM training data are involved. We summarize the card fields adapted from the standard dataset-documentation and model-card practices, scoped to the environment's scale.

Intended use. Pre-deployment stress-testing of LLM agents on the longitudinal workout-recommendation subskill of coaching under wearable-data uncertainty. Suitable uses: model selection and regression testing by research teams and product developers; ablation of tool-use and physiological-reasoning capabilities. Unsuitable uses: clinical decision-making, unsupervised deployment to end users, certification of a model as "safe" for any real population.

Scope exclusions. Pregnancy, active eating-disorder recovery, clinical cardiovascular or metabolic conditions, and users under 18 are excluded by design from the archetype set. Strength training, sport-specific skill development, mixed-modality programming, nutrition and sleep coaching, and non-English prompting are outside the action space. The environment provides no signal for any of these subskills. Perceived-exertion modelling (Borg / Foster session-RPE; interval structure; cumulative-fatigue RPE adjustments) is also out of scope: the compliance model's difficulty scalar is an HR-zone-allocation proxy, not a Borg-compatible RPE measurement.

Authorial provenance of parameters. Simulator parameters fall into two categories.

Literature-anchored constants: Banister $\tau_1 = 45$d, $\tau_2 = 13$d, $K_1 = 0.15$, $\alpha = 0.03$ (Banister et al., 1991; Morton et al., 1990; Busso, 2003; Hellard et al., 2006); ACWR sweet-upper 1.3 and exponential multiplier calibrated to $m = 3$ at ACWR $= 1.8$ (Gabbett, 2016); Tanaka $\text{HR}_{\max}$ formula (Tanaka et al., 2001); injury severity distributions and recovery ranges (per-injury science and medicine cited in the injury registry).

Author-designed calibration choices: logistic-compliance coefficients; SE-gated refusal onset and slope; 0.5 non-preferred structural penalty; Edwards-style zone weights $[0, 1.5, 3.0, 5.0]$; consecutive-high-fatigue threshold (5d) and decay (0.08/d); archetype narratives; and the 9 archetype specifications themselves. These are not transcribed from any source paper; magnitudes are chosen to hit the day-1 compliance targets in Table 2 and to match published adherence-range envelopes.

Released assets.

- **Archetypes**: 9 YAML specifications with physiological, behavioral and environmental parameters (see Appendix A).
- **Mock wearable data generators**: date-seeded observation functions (Appendix D) producing wearable-format responses.
- **Simulation engine**: Banister state machine, ACWR injury model, and logistic compliance dynamics.
- **Reproduction scripts**: regenerate every deterministic number reported in this paper.
- **Evaluated LLM-agent rollouts**: generated with the user's own API keys; not redistributed.

Known failure modes. (1) Deterministic injury is a simplification of multi-factorial real injury risk. (2) Compliance coefficients are calibrated, not fitted to real adherence data. (3) Sim-to-real correspondence of CGR is not empirically validated. (4) Archetype weights are uniform and do not model a real demographic distribution. See §5 for the full list.

Maintenance and community contributions. Archetype YAMLs can be contributed without code changes; accepted archetypes are versioned alongside the environment and subject to community review. Simulator constants are versioned, and paper text is automatically audited against the code on each revision to guard against drift.

O EXTENDED LIMITATIONS AND BROADER IMPACT

This appendix expands the four caveats summarized in §6: cold-start training history, oracle-as-baseline, action-space asymmetry, and deployment-safety considerations.

Oracle as a baseline, not a ceiling. The oracle is a DP heuristic with hidden-state access; it knows the exact compliance function and injury model. It cannot improve compliance through coaching quality and only optimizes within fixed parameters. On low-compliance archetypes (Silent User: $P(\text{easy}) = 43\%$, $P(\text{hard}) \approx 0\%$ in Table 4), even the oracle's achievable gain is modest. The oracle does not bound the value of LLM evaluation in adversarial scenarios: even at similar CGR, the LLM's *behavior* can still be assessed for safety violations such as training through injury signals or ignoring recovery metrics. These qualitative failure modes are invisible to CGR but matter for deployment.

Action-space asymmetry. The agent has no positive "recommend rest" channel: abstaining from `create_workout` is the only way to recommend rest, so a rest decision and inaction look identical. This matches real wearable apps (rest is the absence of a scheduled session), but the benchmark cannot tell deliberate rest from disengagement. The nine archetypes have uniform population weights — the grid is a parameter-space diversity set, not a demographic sample. Richer action dimensions from elite coaching (target pace, interval structure, periodization phase labels, cross-training swaps) are not part of the tool API and are outside FITSIM's scope.

Broader impact and deployment safety. Three caveats matter for deployment: (i) The Banister model is validated in science and medicine on trained athletes at multi-month horizons (Banister et al., 1991; Morton et al., 1990; Busso, 2003; Hellard et al., 2006); its behaviour at recreational year-scale with a pooled TRIMP formulation is an extrapolation, and ranking differences smaller than the $\pm 20\%$ sensitivity bands of Appendix H are not evidence of clinical superiority. (ii) four-zone API does not resolve the Z2/Z3 boundary used in 5-zone polarized programming (Seiler & Kjerland, 2006); FITSIM measures HR-zone distribution shift, not polarization convergence, and a high CGR can coexist with pacing errors invisible at the 4-zone quantization. (iii) CGR > 1 means an agent outperforms the uncoached trajectory on simulated fitness delta. It does not mean the agent is safe for unattended deployment, that it transfers to clinical populations, or that its recommendations meet the informed-consent and medical-device standards a deployed fitness-coaching product must clear. Use FITSIM to compare candidate models during development and triage behaviours for further safety testing, not to certify a model for release.

P EXTENDED RELATED WORK

This appendix surveys persona-LLM simulators and the LLM-role-play calibration-drift literature that motivates our deterministic state machine.

LLM-role-played persona simulators. PatientSim (Kyung et al., 2025) couples MIMIC-ED clinical profiles with LLM persona axes (personality, language proficiency, memory, confusion) for single-encounter ED doctor-LLM evaluation. Related LLM-role-play patient simulators include AgentClinic (Schmidgall et al., 2024), SynthAgent (Aghaee et al., 2026), EvoPatient (Du et al., 2025), MedPI (Diego et al., 2026), and DEPROFILE (Li et al., 2026a); DEPROFILE extends scope to multi-session counseling histories but still without physiological state. Organ-Agents (Chang et al., 2025) simulates cross-organ physiology with per-system LLM agents for ICU digital twins, aimed at clinician decision support rather than coaching. Closest to us in domain, Yun et al. (Yun et al., 2025) synthesise health-condition user profiles with an LLM front-end to evaluate coaching agents; the user persona is still LLM-authored. UserBench (Qian et al., 2025) and SOTOPIA (Zhou et al., 2023) share the LLM-user-simulation framing for social interaction.

LLM-role-play failure modes. A parallel literature shows why LLM-role-played personas are fragile substrates for longitudinal evaluation. Abdulhai et al. (2025) document persistent persona drift in off-the-shelf LLMs simulating a target persona across multiple turns and reduce inconsistency by over 55% via multi-turn RL fine-tuning; SimBench (Hu et al., 2025) finds the best available models reach only $40.8/100$ on human-simulation fidelity across 20 datasets; critiques surface "assistant nature" (Naous et al., 2025), calibration error up to $\sim 20\%$ (Seshadri et al., 2026), and 39% multi-turn coherence degradation (Laban et al., 2025). FITSIM sidesteps these issues by construction: each archetype is a deterministic model anchored in sport science and medicine (Banister physiology, ACWR injury gating, multinomial-logit compliance with archetype-fixed motivation/discipline/trust, and frequency-adapted self-efficacy as the only evolving psych state) rather than an LLM-role-play artifact. The simulator is bit-reproducible under a fixed seed: perturb a single named parameter and the resulting CGR shift is the causal effect of that parameter, with no persona-drift confound.