SIMLIFE: PATTERN UNDERSTANDING FOR LONG-HORIZON HUMAN-AGENT PARTNERSHIP

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding humans over long horizons requires agents to infer not only what people need in the moment, but also how routines form, why they repeat, and when they change. We introduce **SIMLIFE**, a scalable platform for simulating long-term household life with rich visual observations, ground-truth action logs, and synthetic dialogues with audio. Built on **SIMLIFE**, **SIMLIFE-BP** evaluates long-context pattern understanding: the ability to infer latent behavioral rules from weeks or months of everyday observations. The benchmark contains 106 episodes averaging 15.49 hours and 38.57 in-game days, and 1,439 question-answer pairs. Each task probes direct, counterfactual, noisy, and inverse reasoning under different levels of rule hints. Evaluating frontier models and architectures, we find that current models often achieve surface-level prediction without comprehensive rule understanding, rely on frequency-based heuristics rather than if-then reasoning over evidence, and struggle to adapt when behavioral patterns change. These findings suggest that long-context pattern understanding remains a major bottleneck for future embodied agents, while **SIMLIFE** opens a broader space for studying memory, personalization, adaptation, and planning in everyday human-AI interaction.

1 INTRODUCTION

Understanding humans' implicit intents and motives has long been a fundamental challenge. In long-context settings, this challenge goes beyond interpreting a person's immediate needs or beliefs (Ma et al., 2023; Peng et al., 2025): an agent must also recognize recurring behaviors and infer the underlying patterns that govern them over time. Such longitudinal understanding is essential for agents that continuously interact with people across weeks or months of everyday life.

Long-term behavioral patterns cannot be understood from isolated observations alone. For example, inferring that *Bob drinks coffee every weekday morning or after staying up late* requires an agent to form hypotheses (e.g., "Bob may drink coffee every day."), revisit relevant observations, and repeatedly verify or revise them as evidence accumulates. This makes pattern understanding a genuinely long-horizon reasoning problem, beyond retrieving and using a previously observed fact.

To evaluate this form of long-context understanding under controlled yet realistic conditions, we need simulation environments that support extended timelines and rich everyday behaviors. Existing embodied simulation platforms provide valuable testbeds for embodied tasks (Puig et al., 2018; Shridhar et al., 2020; Puig et al., 2024; Ren et al., 2026; Zhou et al., 2026), but long-term household routines with diverse contextual cues remain difficult to simulate at scale. To address this gap, we propose **SIMLIFE**, a scalable platform for simulating long-term everyday life in a home-like environment. Built on *The Sims 4*, **SIMLIFE** supports scripted simulations of multi-character household activities with rich visual observations, annotations, and diverse event categories (Figure 1). We use this platform as a controllable foundation for constructing long-context benchmarks grounded in everyday behavior.

In this paper, we instantiate **SIMLIFE** as **SIMLIFE-BP**, a benchmark for behavioral pattern (BP) understanding: the ability to infer behavioral regularities, the rules behind them, and how these patterns evolve over time. **SIMLIFE-BP** contains 106 ultra-long episodes spanning up to 48.23 hours and 120 in-game days, with 1,439 questions in total. Each episode follows a controlled, deterministic latent behavioral pattern governed by temporal or event-based rules, together with rich synchronized

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.

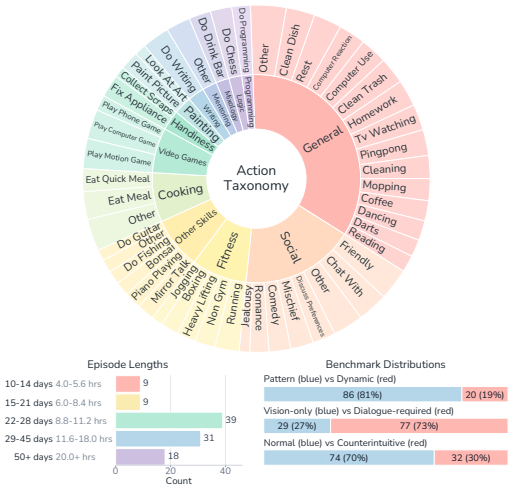

Figure 1: **SIMLIFE-BP** distribution overview. **Top:** action taxonomy with log-scaled counts. **Bottom-left:** episode lengths distribution, where one in-game day corresponds to 24 real minutes, or 0.4 real hours. **Bottom-right:** composition over three dimensions.

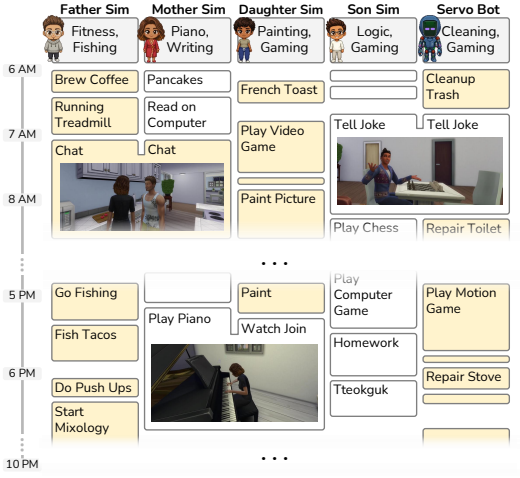

Figure 2: Example simulated day in **SIMLIFE**. Each column shows the 6 AM–10 PM schedule of one Sim, whose behavioral preferences are annotated above the timeline. Interactions between Sims are shown as shared blocks. Some time slots are collapsed for presentation.

annotations for contextual grounding. To systematically evaluate pattern understanding, **SIMLIFE-BP** combines *four question types* that comprehensively probe pattern inference and application with *three hint levels* that vary the amount of prior rule information. This design tests whether models merely recognize surface regularities or can infer and robustly apply the underlying behavioral rules across diverse contexts.

Our evaluation reveals a consistent picture. Models improve substantially as evidence becomes cleaner and more structured, yet even under favorable conditions, latent rule inference remains a major bottleneck. Our comprehensive question design further exposes limited robustness: models often succeed on easier forward predictions but fail to apply the same inferred pattern consistently across reasoning variants. Looking beyond final-task performance, we find that models frequently fall back on simple heuristics such as frequency counting rather than systematic if-then reasoning, and struggle to revise their understanding when the underlying pattern itself changes. Together, these results suggest that long-context pattern understanding requires more than remembering distant observations: agents must continually form, test, and revise hypotheses about behavior over time. More broadly, **SIMLIFE** provides a foundation for studying this kind of persistent reasoning in long-horizon human-AI interaction, toward embodied agents that can build and adapt their understanding throughout extended periods of everyday life.

2 RELATED WORK

Long Context Learning Long-context learning has long been studied across application domains, from early work on document summarization and long-document understanding (Luhn, 1958; Edmundson, 1969; Salton et al., 1994) to modern long-context understanding (Beltagy et al., 2020; Liu et al., 2024; Kamradt, 2023; Jelassi et al., 2024), dialogue and chat history modeling (Bei et al., 2026), agentic tool use (Yao et al., 2023; Acharya et al., 2025; Fu et al., 2024), multimodal understanding (Long et al., 2026; Chen et al., 2026; Yan et al., 2025a), and embodied AI with persistent memory (Wang et al., 2024; Dai et al., 2026). Architecturally, this progress has been supported by increasingly large context windows (Reid et al., 2024; Bai et al., 2025a; Peng et al., 2024; Gemma Team et al., 2026), test-time adaptation (Ma et al., 2026; Zhang et al., 2025), continual learning (Deng et al., 2025), and self-evolving agents (Gao et al., 2026). Everyday assistants fall into this setting, yet the growing scale and complexity of long-context benchmarks raise a key question: what makes a task uniquely long-context, beyond simply requiring longer inputs? Echoing Hsieh et al. (2024); Bai et al. (2024; 2025b), **SIMLIFE-BP** addresses this question by targeting intrinsically long-horizon reasoning

rather than local retrieval over chunked segments, requiring models to integrate and cross-reference evidence scattered across days, weeks, and modalities to infer latent behavioral patterns.

Long Video Understanding Long video understanding benchmarks have expanded across duration and modality, covering hour-long QA (Wu et al., 2024), extreme-length comprehension (Wang et al., 2025a), and egocentric scenarios (Grauman et al., 2022; Mangalam et al., 2023; Yang et al., 2025; Long et al., 2026; Chen et al., 2026; Yan et al., 2025a). Yet many still operationalize long-video understanding as retrieval or localization. **SIMLIFE-BP** instead targets intrinsically cross-temporal understanding, where answers require integrating implicit evidence across multiple days, modalities, and interactions. By combining continuous daily-life simulations with controlled scripts and sampled agent profiles, **SIMLIFE-BP** enables controllable evaluation of latent behavioral pattern inference.

Pattern Recognition Pattern recognition spans perceptual regularity detection, sequential pattern mining, and abstract rule discovery (Bongard, 1970; Klahr & Dunbar, 1988; Josephson & Josephson, 1996; Agrawal & Srikant, 1995; Srikant & Agrawal, 1996; Mannila et al., 1997; Pei et al., 2001). Recent LLM benchmarks test rule induction from compact visual puzzles, symbolic examples, noisy observations, or programmatic input-output pairs (Chollet, 2019; Chollet et al., 2025; Hua et al., 2025; Li et al., 2025a;b). Long-context variants (Yan et al., 2025b; Kuratov et al., 2024) require aggregating many examples or scattered facts, but still use structured textual inputs. In contrast, **SIMLIFE-BP** evaluates behavioral pattern recognition from unstructured ultra-long multimodal observations, requiring models to infer reusable latent rules from sparse, perceptually grounded, temporally distributed evidence.

3 SIMLIFE BENCHMARK

3.1 PLATFORM & DATASET

SIMLIFE is built on *The Sims 4*, where we simulate multi-character household activities observed from the egocentric perspective of a serving robot called *Servo Bot*. Each simulated day spans 6 AM to 10 PM in-game, corresponding to approximately 24 minutes in real time. Figure 2 illustrates one example simulated day. Characters (or Sims) are initialized with behavioral profiles specifying probabilistic activity preferences, which shape their daily routines and interactions. We then sample 466 in-game simulated days as the building blocks of our dataset. Each ultra-long video instance is constructed by concatenating a sequence of these day-level clips. We use a full simulated day as the basic unit because it forms a complete behavioral cycle, allowing clips to be recombined without introducing inconsistencies within a character's daily behavior.

Each recorded day is further enriched with fine-grained, synchronized annotations. We collect ground-truth game logs filtered to Servo Bot's field of view and introduce context-oriented dialogues, ranging from natural everyday conversations to utterances that reference off-screen events or evolving situations. These dialogues are further rendered as spatialized conversational audio using Sim-specific voices, allowing the observer to distinguish both who is speaking and the speaker's relative distance. Together, these components provide rich, temporally aligned observations for studying long-horizon understanding in simulated daily life.

3.2 BENCHMARK DESIGN

Pattern Definition We formulate the task as inferring a simulated character's behavioral pattern from ultra-long observations. Each pattern specifies whether and when a character performs a target activity under a set of temporal or event-based conditions. For example, the pattern *"Bob drinks coffee on weekdays or after staying up late"* links coffee drinking to the day of the week and a prior-night event. To instantiate such a pattern, we filter and link simulated days into a long video sequence that repeatedly exhibits evidence of the underlying rule. The model must aggregate distributed observations to recover the rule governing the character's behavior. See Appendix B for details.

Our simulated patterns are controlled and deterministic once their underlying conditions are met. Under the rule above, for example, Bob drinks coffee on every weekday and always drinks coffee after staying up late, without randomly skipping the activity when either condition holds. Conditioning events can themselves occur stochastically at an independently specified frequency, but once such an event occurs, its effect on the target behavior is deterministic.

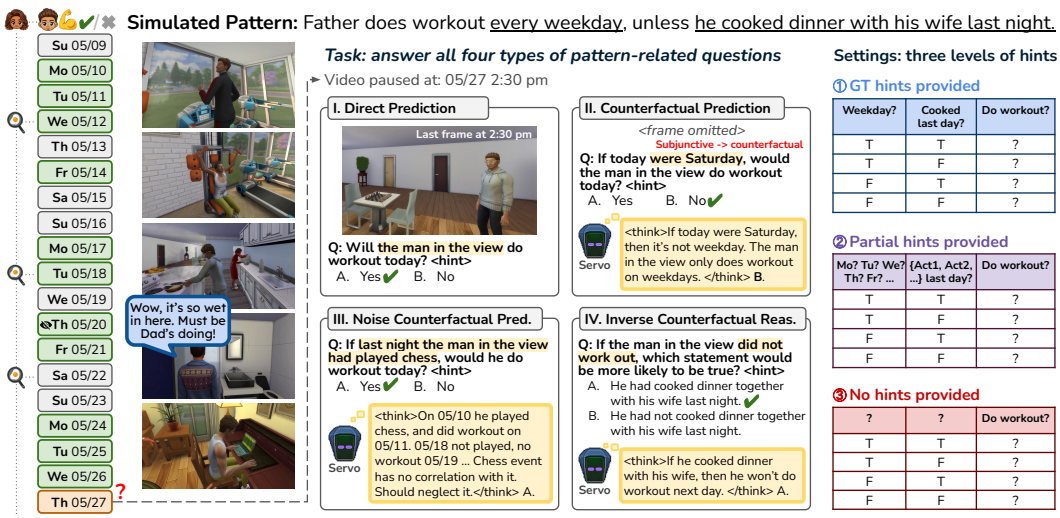

Figure 3: One example task of **SIMLIFE-BP** benchmark. Given the Father Sim's workout routine (shown at the top), we simulate three weeks of daily life that reflect this underlying pattern, recorded from the egocentric view of the Servo Bot. After sufficient observation, models are evaluated using four types of questions designed to assess their consistency in pattern understanding. We further provide three levels of hints that systematically ablate the skills required for long-term reasoning.

Note that behavioral patterns can be represented in many different forms, including graphs, procedures, policies, and probabilistic models. Within the scope of this work, we represent patterns using symbolic rules in the form of if–then propositions (e.g., if X and Y, then A). We choose this representation because such rules are highly interpretable and provide an explicit characterization of the agent's understanding of behavioral regularities. We view this formulation as an initial step rather than a restriction of the framework. Building on our simulation platform and data-generation pipeline, future work can explore alternative and potentially richer representations of behavioral patterns.

Task Design We evaluate models using grouped question answering in a streaming setting. After exposing the model to a sufficiently long video history of in-house activities, we pause the video and instantiate a **task**, defined as a group of four binary multiple-choice questions[1] about the behavioral pattern at that point. The four questions probe complementary forms of pattern understanding:

1. **Direct Prediction:** Predict the most likely behavior given the current conditions.
2. **Counterfactual Prediction:** Predict the most likely behavior under a hypothetical change in conditions.
3. **Noisy Counterfactual Prediction:** Predict the most likely behavior under hypothetical conditions that include irrelevant information.
4. **Inverse Counterfactual Reasoning:** Infer the most plausible condition or cause given a counterfactual behavior.

Each behavioral pattern yields multiple tasks, inserted at query points chosen to cover different combinations of the pattern conditions. In the coffee example, we query the model on days corresponding to different combinations of weekday/weekend and with/without stayed-up-late. This allows us to test whether the model has learned the complete rule rather than a partial heuristic.

In addition, we evaluate each task under three hint levels that vary how much rule information is provided. In **Full Hints**, the relevant conditions are directly specified, e.g., *"the activity is related to whether it is a weekday"*. In **Partial Hints**, only the abstract type of each condition is revealed, e.g., *"the activity is related to the day of the week"*; this yields a closed hypothesis space while still requiring models to eliminate inconsistent rules from evidence. In **No Hints**, the model must infer the pattern entirely from observation, which is the most realistic, but unbounded setting (see Appendix C).

Finally, to avoid asking questions before sufficient evidence is available, we use the Partial Hints setting as a reference point for query timing. For example, if *Bob plays ping-pong every Monday*

[1] In special cases (`Hist` task in Section 4.3), there are no direct prediction questions.

and Friday and Partial Hints indicate that the pattern depends on the day of the week, observing one complete week is sufficient to determine his future behavior under our deterministic assumption. We call the earliest such point the **convergence point**.

For more complex conditions and different starting dates, the convergence point is less obvious. We therefore run a hypothesis-elimination procedure, starting from all candidate rules expressible over the condition space specified by Partial Hints and removing hypotheses inconsistent with each newly observed day. Formally, we define the convergence point as the earliest day after which all remaining hypotheses produce predictions consistent with the ground-truth rule for every subsequent day, and insert tasks only after this point. This ensures that each task is, in principle, answerable from the accumulated evidence with an unambiguous ground-truth prediction under the Partial Hints and Full Hints settings. While No Hints more closely reflects a realistic setting, it does not expose the ground-truth condition space required to define such a convergence point. We therefore view deterministic pattern understanding as a controlled prerequisite for studying more dynamic and probabilistic behavioral patterns. More details are provided in Appendix D.

Benchmark Composition As is shown in Figure 1, **SIMLIFE-BP** contains 86 episodes for *pattern understanding* and 20 episodes for *dynamic understanding*. Pattern understanding episodes involve a single behavioral pattern, while dynamic understanding episodes involve multiple behavioral patterns and require models to identify distinct patterns as well as detect pattern shifts over time. Among them, 29 are solvable from vision alone, while 77 additionally require dialogue (audio).

To further probe contextual reasoning, 30% of episodes follow *counterintuitive patterns*, which associate activities with unusual or implausible conditions that cannot be inferred through common sense alone. For example, the pattern *"drink coffee after playing the piano"* is individual-specific and unlikely to arise from prior general knowledge captured by large models. Models must therefore rely primarily on evidence from the video context rather than world priors when answering questions.

4 EXPERIMENTS & RESULTS

We evaluate AI models' ability to understand behavioral patterns from ultra-long observations along three complementary dimensions: **whether models can infer such patterns, how their understanding develops as evidence accumulates, and how they adapt when patterns evolve.**

4.1 RQ1: CAN AI MODELS INFER PATTERNS FROM ULTRA-LONG OBSERVATIONS?

We first assess the overall capability of models to recognize and reason about behavioral patterns accumulated over extended observation periods.

4.1.1 EXPERIMENTAL SETUP

We designed experiments varying both the input modality and the degree of provided context. We evaluate models under four input conditions:

- **Vision + Language (V+L):** Video frames (without audio) and dialogue transcripts.
- **Vision + Audio (V+A):** Video frames plus raw audio recordings of dialogue instead of transcripts.
- **Language Only (L):** Event logs (action & dialogue) concatenated into a single input. This bypasses the vision process and uses ground-truth activities instead.
- **Language Only, noiseless (L-N):** Filtered logs with main and directly related events.

Model Choice We evaluate a broad set of models, including the Qwen3-VL family (4B, 8B, and 32B) (Bai et al., 2025a) with Chain-of-Thought (CoT) prompting; reasoning models with hybrid attention such as the Qwen3.5 (Qwen Team, 2026) and Gemma4 (Gemma Team et al., 2026) families, covering both dense and mixture-of-experts (MoE) variants; Qwen3-Omni (Xu et al., 2025) with an audio channel, and the Gemini (Google, 2025; Comanici et al., 2025) family as representative proprietary models. For vision-based input conditions, we follow the Socratic Models framework (Zeng et al., 2023): we divide each day's video into four clips of approximately six minutes each, generate a summary for each clip, and concatenate all summaries with the task questions. For language-only variants, we concatenate the logs and questions into a single input pass. Implementation details are provided in Appendices J and K.

Table 1: Main results on **SIMLIFE-BP** (381 tasks / 1,439 questions per hint level). We report All-Correct rate (AC) and question-level accuracy (Acc) with 95% binomial CI half-widths. Each Sub Acc cell shows per-question-type accuracy, left-to-right: Direct, Counterfactual, Noise Counterfactual, Inverse Counterfactual; the gray line marks chance. Shading highlights within-row best variants, and **bold** marks the overall best in each column.

Model	FPS	Thinking	Full Hints			Partial Hints			No Hints		
			AC	Acc	Sub Acc	AC	Acc	Sub Acc	AC	Acc	Sub Acc
Vision + Language (V+L)											
Qwen3-VL-4B-Instruct	0.5	CoT	$10.5^{\pm 3.1}$	$49.8^{\pm 2.6}$		$11.5^{\pm 3.2}$	$50.0^{\pm 2.6}$		$9.4^{\pm 2.9}$	$50.3^{\pm 2.6}$	
Qwen3-VL-8B-Instruct	0.5	CoT	$12.6^{\pm 3.3}$	$50.1^{\pm 2.6}$		$9.7^{\pm 3.0}$	$48.6^{\pm 2.6}$		$7.6^{\pm 2.7}$	$48.0^{\pm 2.6}$	
Qwen3-VL-32B-Instruct	0.5	CoT	$11.0^{\pm 3.1}$	$51.9^{\pm 2.6}$		$13.6^{\pm 3.4}$	$53.2^{\pm 2.6}$		$12.1^{\pm 3.3}$	$53.0^{\pm 2.6}$	
Qwen3.5-9B	0.5	R	$12.3^{\pm 3.3}$	$48.9^{\pm 2.6}$		$14.2^{\pm 3.5}$	$51.4^{\pm 2.6}$		$13.4^{\pm 3.4}$	$51.0^{\pm 2.6}$	
Qwen3.5-35B-A3B	0.5	R	$13.1^{\pm 3.4}$	$52.2^{\pm 2.6}$		$14.2^{\pm 3.5}$	$53.3^{\pm 2.6}$		$12.3^{\pm 3.3}$	$52.5^{\pm 2.6}$	
Gemma-4-26B-A4B-it	0.5	R	$7.6^{\pm 2.7}$	$48.8^{\pm 2.6}$		$7.6^{\pm 2.7}$	$48.1^{\pm 2.6}$		$10.8^{\pm 3.1}$	$49.3^{\pm 2.6}$	
Gemini-3-flash-preview	0.5	R	$14.7^{\pm 3.6}$	$53.0^{\pm 2.6}$		$13.9^{\pm 3.5}$	$54.1^{\pm 2.6}$		$14.2^{\pm 3.5}$	$54.8^{\pm 2.6}$	
VideoTree (Mixtral-8x7B)	1	CoT	$11.0^{\pm 3.1}$	$54.0^{\pm 2.6}$		$9.7^{\pm 3.0}$	$51.7^{\pm 2.6}$		$10.8^{\pm 3.1}$	$52.6^{\pm 2.6}$	
Vision + Audio (V+A)											
Qwen3-Omni-30B-A3B + Qwen3.5-9B	0.5	R	$10.8^{\pm 3.1}$	$48.1^{\pm 2.6}$		$11.0^{\pm 3.1}$	$48.6^{\pm 2.6}$		$10.8^{\pm 3.1}$	$48.6^{\pm 2.6}$	
M3-Agent	1	R	$10.0^{\pm 3.0}$	$53.4^{\pm 2.6}$		$10.2^{\pm 3.0}$	$55.0^{\pm 2.6}$		$12.1^{\pm 3.3}$	$54.6^{\pm 2.6}$	
Language Only (L)											
Qwen3.5-9B	–	R	$9.2^{\pm 2.9}$	$50.9^{\pm 2.6}$		$13.6^{\pm 3.4}$	$54.6^{\pm 2.6}$		$14.2^{\pm 3.5}$	$54.6^{\pm 2.6}$	
Qwen3.5-35B-A3B	–	R	$18.1^{\pm 3.9}$	$61.1^{\pm 2.5}$		$18.4^{\pm 3.9}$	$59.3^{\pm 2.5}$		$16.0^{\pm 3.7}$	$58.7^{\pm 2.5}$	
Gemma-4-26B-A4B-it	–	R	$11.3^{\pm 3.2}$	$53.9^{\pm 2.6}$		$11.0^{\pm 3.1}$	$54.4^{\pm 2.6}$		$11.3^{\pm 3.2}$	$54.8^{\pm 2.6}$	
Gemini-2.5-flash	–	R	$17.9^{\pm 3.8}$	$58.6^{\pm 2.6}$		$21.0^{\pm 4.1}$	$60.2^{\pm 2.5}$		$16.3^{\pm 3.7}$	$56.4^{\pm 2.6}$	
Gemini-3-flash-preview	–	R	$27.8^{\pm 4.5}$	$69.2^{\pm 2.4}$		$\mathbf{33.3}^{\pm 4.7}$	$69.4^{\pm 2.4}$		$27.8^{\pm 4.5}$	$66.5^{\pm 2.4}$	
Gemini-3.1-flash-lite	–	R	$20.7^{\pm 4.1}$	$59.9^{\pm 2.5}$		$21.0^{\pm 4.1}$	$61.2^{\pm 2.5}$		$18.6^{\pm 3.9}$	$59.1^{\pm 2.5}$	
Language only, noiseless (L-N)											
Qwen3.5-9B	–	R	$23.6^{\pm 4.3}$	$65.5^{\pm 2.5}$		$18.4^{\pm 3.9}$	$61.2^{\pm 2.5}$		$16.0^{\pm 3.7}$	$56.5^{\pm 2.6}$	
Qwen3.5-35B-A3B	–	R	$34.9^{\pm 4.8}$	$73.0^{\pm 2.3}$		$32.0^{\pm 4.7}$	$\mathbf{70.3}^{\pm 2.4}$		$21.5^{\pm 4.1}$	$62.7^{\pm 2.5}$	
Gemma-4-26B-A4B-it	–	R	$30.2^{\pm 4.6}$	$72.1^{\pm 2.3}$		$26.0^{\pm 4.4}$	$68.4^{\pm 2.4}$		$22.3^{\pm 4.2}$	$64.1^{\pm 2.5}$	
Gemma-4-31B-it	–	R	$\mathbf{36.2}^{\pm 4.8}$	$\mathbf{74.6}^{\pm 2.2}$		$30.2^{\pm 4.6}$	$69.6^{\pm 2.4}$		$\mathbf{29.4}^{\pm 4.6}$	$\mathbf{67.6}^{\pm 2.4}$	
Gemini-2.5-flash	–	R	$31.5^{\pm 4.7}$	$71.8^{\pm 2.3}$		$22.3^{\pm 4.2}$	$65.3^{\pm 2.5}$		$22.1^{\pm 4.2}$	$63.7^{\pm 2.5}$	
Gemini-3-flash-preview	–	R	$24.1^{\pm 4.3}$	$68.2^{\pm 2.4}$		$24.9^{\pm 4.3}$	$67.3^{\pm 2.4}$		$23.1^{\pm 4.2}$	$63.9^{\pm 2.5}$	

We also include representative agentic baselines designed for temporal search, i.e. VideoTree (Wang et al., 2025b), and memory management, i.e. M3-Agent (Long et al., 2026).

Hint Levels We evaluate models with three hint levels as introduced in Section 3.2.

Metrics We report three primary metrics: *All-Correct Rate* (AC), the proportion of tasks for which the model answers all sub-questions correctly, reflecting the consistency and completeness of its pattern understanding; *Average Accuracy Rate* (Acc), the mean accuracy across all sub-questions within a task, as well as *Sub-questions Accuracy* (Sub Acc), providing a finer-grained measure.

4.1.2 RESULTS

Evidence Quality Drives Performance Table 1 shows an overall trend of improved performance as inputs become cleaner and more structured. In the V+L setting, most models remain near chance on Acc, with only Gemini 3 Flash consistently performing above chance across hint levels. In contrast, models consistently perform above chance in L-N, where Gemma-4-31B-it reaches 36.2% AC and 74.6% Acc with Full Hints. The substantial improvement from V+L to L-N highlights the difficulty of extracting relevant evidence from long multimodal observations, while the remaining gap even with oracle-filtered inputs indicates that latent rule inference itself remains challenging.

Structured Reasoning Emerges in L and L-N Across hint levels, V+L shows no consistent benefit from stronger hints, while a signal of structured reasoning begins to emerge in L and becomes clearest in L-N, where Full Hints generally perform best. Reasoning-trace analysis helps explain this progression: Full Hints direct models toward the relevant variables, but using them effectively depends on whether the corresponding evidence can be recovered. In V+L, extracting the deciding evidence from long visual observations is already challenging, and it may be absent from the generated diaries; in L, the evidence is available but can be buried in noisy logs; in L-N, it is both complete and structured, allowing the hints to translate more reliably into correct rule use. Together, these results suggest that structured pattern reasoning begins to emerge once relevant evidence becomes accessible, but remains strongly constrained by evidence extraction and organization.

All-Correct Rate (AC) Guards Against Shortcutting The per-question-type breakdown (Sub Acc) reveals a consistent asymmetry: normal counterfactual questions (CF) are much easier than

inverse (IC) and noise-injected (NC) variants, and this gap widens as inputs become cleaner: under No Hints, the median CF-IC gap increases from +4.2% in V+L to +16.2% in L and +32.1% in L-N, while IC accuracy changes little. In addition, although Gemma-4-31B-it (L-N) achieves 86.6% accuracy on CF questions, only 36.2% of tasks are solved consistently across all question variants. These results support the design of AC, which requires models to answer all variants consistently and therefore prevents high performance on easier forward questions alone from being mistaken for genuine pattern understanding. In this way, AC better distinguishes systematic rule inference from answers obtained through question-specific shortcuts, common-sense priors, or chance. We show the full results in Table 21.

4.2 RQ2: How do AI models develop pattern understanding over time?

Having examined models' overall pattern-understanding performance, we next investigate how this understanding develops as observations accumulate over time, and whether their evolving beliefs align with human judgments. Instead of focusing only on the final prediction, we probe how models incrementally update their beliefs about the underlying pattern across successive observations.

4.2.1 Experimental Setup

We evaluate four representative models (Gemini-2.5-flash, Gemini-3-flash-preview, Gemma-4-26B-A4B-it, and Qwen3.5-35B-A3B) under the *Language Only*, *Partial Hints* setting. For each day d in an episode, the model predicts whether the target activity will occur, yielding a daily belief trajectory $P_d(\text{yes})$. For models supporting token log-probabilities, we derive $P_d(\text{yes})$ by summing and normalizing the probability mass of *yes*/*no* variants at the answer token; for Gemini-3-flash-preview, which does not expose log-probabilities, we map its emitted yes/no answer to 1/0. Rather than evaluating only the final prediction, this day-by-day trajectory captures how a model's belief shifts from uncertainty to confidence as evidence accumulates.

Since a model's internal decision process is not directly observable, we compare its trajectory against interpretable reference baselines as behavioral probes: if a model's predictions closely track a baseline, this suggests that it may be following a similar decision strategy. We consider five reference baselines. Two are shallow heuristics that do not perform rule inference: (B1) *frequency-based*, predicting from the historical base rate, and (B2) *last-day-based*, extrapolating from the previous day's outcome. Three are advanced references: (B3) *ground-truth* oracle, (B4) *human predictions* (Appendix G), and (B5) *hypothesis-elimination*, which prunes candidate rules against accumulated evidence (Appendix D).

Metrics We analyze the models' daily predicted probabilities for yes/no answers. We use the **Brier score** (Brier, 1950) to measure the discrepancy between predicted probabilities $P_d(\text{yes})$ and ground-truth outcomes y_d, computed as

$$\text{Brier} = \frac{1}{D}\sum_{d=1}^{D}\left(P_d(\text{yes}) - y_d\right)^2,$$

where $y_d \in \{0, 1\}$; lower scores indicate better probabilistic predictions. Accuracy (Acc), balanced accuracy (B-Acc), positive F1-score (F1^+), and averaged $P(\text{yes})$ ($\bar{p}_{\text{yes}}$) are also reported for reference. In addition, we report Spearman correlation between each model's daily prediction trajectory and the five reference baselines described above.

4.2.2 Results

Table 2 reports the pairwise Spearman correlations among the $P_d(\text{yes})$ trajectories across all sources, measuring how similarly each trajectory evolves over the chain. For Qwen, Gemma, and Gemini 2.5 Flash, the strongest non-self correlation is with the frequency-based baseline (28.8%, 32.2%, 38.5% respectively). These results suggest that, given partial hints, models tend to **default to simple heuristics, like counting instances of the target activity, rather than evaluating candidate rules against the accumulated evidence.**

By construction, the hypothesis-elimination baseline captures a sequential process of pruning candidate rules against observed evidence. Human predictions are strongly correlated with this baseline (78.1%), suggesting that it captures an important aspect of how humans update their beliefs over

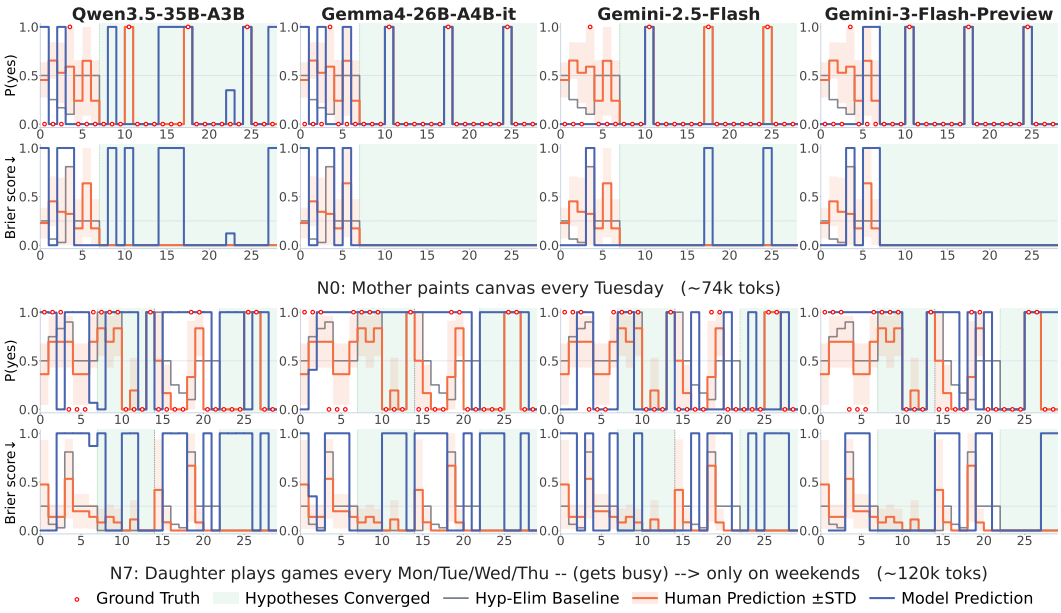

Figure 4: Per-day trajectories of model predictions and human judgments. Bottom rows show the per-day Brier score $(P_d(\text{yes}) - y_d)^2$. A rapid convergence of Brier score of model prediction toward zero after hypothesis elimination indicates successful pattern understanding and retention.

Table 2: Spearman rank correlation of $P_d(\text{yes})$ trajectories (%). Columns [1]–[9] correspond to the indexed source rows in the leftmost column. Blue/light blue: best/second-best non-self correlation per row ([1]–[5]). Orange/light orange: best/second-best per column ([5], Acc, B-Acc, F1$^+$, Brier). $\bar{p}_{\text{yes}}$

Source	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	n	Acc	B-Acc	F1$^+$	Brier↓	$\bar{p}_{\text{yes}}$
Language Only (L)															
[1] Qwen3.5-35B-A3B	100.0	22.0	20.4	21.4	15.2	28.8	26.2	19.6	23.7	4067	59.7	61.3	46.1	39.8	48.1
[2] Gemma-4-26B-A4B-it	22.0	100.0	18.8	31.4	22.0	32.2	12.8	23.6	25.0	4017	64.0	64.0	48.7	36.0	43.5
[3] Gemini-2.5-flash	20.4	18.8	100.0	21.4	21.7	38.5	22.1	21.4	29.2	4068	71.6	59.6	39.0	28.4	20.1
[4] Gemini-3-flash-preview	21.4	31.4	21.4	100.0	46.7	30.4	9.6	38.8	40.5	4053	74.0	70.7	56.5	26.0	33.4
Human Study															
[5] Human-prediction	15.2	22.0	21.7	46.7	100.0	17.6	4.8	74.3	78.1	194×3	84.5	83.3	79.2	9.8	39.4
Reference Baselines															
[6] Frequency-based	28.8	32.2	38.5	30.4	17.6	100.0	52.0	37.0	50.2	4068	73.0	63.3	45.6	17.0	29.2
[7] Last-day-based	26.2	12.8	22.1	9.6	4.8	52.0	100.0	12.7	19.3	4068	65.3	56.5	36.7	33.7	26.9
[8] Ground-truth-based	19.6	23.6	21.4	38.8	74.3	37.0	12.7	100.0	79.2	4068	100.0	100.0	100.0	0.0	26.6
[9] Hypothesis-elimination-based	23.7	25.0	29.2	40.5	78.1	50.2	19.3	79.2	100.0	4068	88.7	88.0	80.2	6.5	27.7

time. Against this reference, Gemini 3 Flash stands out: its closest non-self baseline is hypothesis elimination (40.5%), and it also shows the strongest alignment with human predictions (46.7%). Together, these results suggest that **Gemini 3 Flash more closely tracks hypothesis-elimination behavior and produces more human-like trajectories over the observation chain**. In contrast, the other models appear more strongly anchored by frequency and recency heuristics, with weaker evidence of systematic hypothesis elimination.

Representative Examples Figure 4 unpacks the aggregate correlations on two chains selected from the human study (see Appendix G). The first case N0 (*"Mother paints every Tuesday"*) is the easiest pattern that hypothesis-elimination resolves within the first week (green area). Both Gemini 3 Flash and Gemma stay close to the human curve here, while Gemini 2.5 Flash picks up the signal early on but reverts to "no" for the end of the chain.

The second case, N7 (*"Daughter plays games every Mon/Tue/Wed/Thu, but then plays only on weekends after she gets busy"*), contains a pattern shift at day 14 (as the pivot). Humans and Gemini 3 Flash adapt quickly: Brier scores on post-pivot weekdays initially spike due to the old pattern, but they return to near zero within a week. In contrast, Gemini 2.5 Flash, Gemma, and Qwen remain locked onto the original rule and continue predicting "yes" on weekdays for the rest of the chain.[2]

[2]We note one fairness asymmetry: humans receive an explicit notice ("the behavior pattern may change after this point"), whereas models must infer the shift through dialogue, thus the post-pivot gap reflects both the scaffolding difference and a failure to recognize that dialogue indicates a change.

Overall, Gemini 3 Flash is the one model whose reasoning traces consistently show explicit cycle detection and post-pivot dialogue use, matching its stronger correlation with both hypothesis-elimination baseline and human prediction.

4.3 RQ3: How do AI models adapt to evolving patterns?

Having examined how models update their understanding as evidence accumulates, we next ask how they respond when the **underlying pattern itself changes**. We examine whether models can detect such changes, adapt their understanding accordingly, and keep knowledge of prior patterns without forgetting.

4.3.1 Experimental Setup

We evaluate dynamic understanding across 20 episodes, each consisting of two sequential behavioral patterns, using three task types: `P1`, questions about Pattern 1 while active; `P2`, questions about Pattern 2 after pattern shift and sufficient exposure to the new pattern; and `Hist`, questions about the earlier Pattern 1 after observing Pattern 2. Representative results appear in Table 3, using `Acc`.

4.3.2 Results

Under L and L-N settings, `P2` is consistently weakest; three of the four L/L-N settings follow `P1` > `Hist` > `P2`. The `P1`→`P2` gap is significant for every model, reaching +25.6% ($p < .001$) for Gemini 3 Flash, showing that **models can identify a schedule, but struggle to update once it changes**. The limitation is online integration of the new pattern. This gap shrinks but does not close for stronger models (see Table 20).

Table 3: Pattern adaptation of representative baselines (partial-hint, question-level accuracy in %). Stars indicate task-level Mann-Whitney significance ($^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$). See full results in Table 20.

	Model	P1	P2	Hist	P1-P2	Hist-P1	Hist-P2
L	Qwen3.5-35B-A3B	63.4	48.7	65.3	+14.8*	+1.8	+16.6*
	Gemini-3-flash	86.9	61.3	74.5	+25.6***	-12.4**	+13.2*
L-N	Qwen3.5-35B-A3B	85.5	65.3	77.3	+20.2***	-8.2**	+12.0
	Gemini-3-flash	79.3	65.3	76.4	+14.0*	-2.9	+11.1
V/Agent	Gemini-3-flash (V+L)	57.9	56.7	55.6	+1.3	-2.4	-1.1
	VideoTree (retrieval)	48.3	51.3	59.3	-3.1	+11.0	+7.9
	M3-Agent (memory)	60.7	56.7	57.9	+4.0	-2.8	+1.2

`Hist` Retrieval Shortcut Notably, we expected `Hist` to be the hardest setting with the worst performance, because it requires retaining the earlier pattern while resisting interference from the current one; yet models generally perform better on `Hist` than `P2`. We attribute this to a shortcut our design intended to avoid but that persists in full-context models: because pre-switch evidence remains verbatim in the context, models can answer `Hist` through local pattern-matching rather than by reasoning about the pattern transition. Accuracy is consistent with this reading: `Hist` sits far above `P2`, yet stays below the clean `P1` (-12.4%, $p < .01$ for Gemini 3 Flash, L) condition, likely because post-shift content dilutes but does not prevent retrieval. Temporal anchors were intentionally vague (e.g., "roughly one month ago") to discourage this shortcut, but wording had no effect on accuracy.

5 Conclusion & Future Work

We presented **SimLife**, a platform for simulating long-term everyday life, and **SimLife-BP**, a benchmark for evaluating pattern understanding from ultra-long multimodal observations. Our results show that current models often rely on surface heuristics, fail to apply inferred rules robustly and struggle to adapt to pattern shifts. These findings highlight long-context pattern understanding as an open challenge for embodied AI agents and **SimLife** provides a foundation for studying memory, personalization, adaptation, and long-horizon planning in everyday human-AI interaction.

It is true that human behaviors are often probabilistic and may exhibit substantial variability and exceptions. However, as the goal of our research is to evaluate models' ability to infer behavioral patterns from observations, we opt to use deterministic rules to provide a well-defined ground truth for systematic and controlled evaluation. This design allows us to isolate the model's ability to discover the underlying behavioral regularities without introducing additional uncertainty about the behaviors themselves. We view this as a controlled starting point. Extending the framework to probabilistic patterns and behaviors with exceptions represents an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets as part of our benchmark (text-to-audio). We have not used generative AI tools for: *"design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results."*, and *"help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses"* are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code, summarizing existing literature, brainstorming, and paper editing for better readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human-study Ethics An institutional review board determined that this study was exempt from ongoing review. We recruited 12 adult English-reading participants through personal outreach, and compensated each participant $20 for a study expected to take approximately one hour, following institutional participant-compensation guidelines. Participants were shown a consent statement describing the study purpose, duration, task format, recorded data, and research use. Only task responses, including symbolic inputs, slider responses, and text-based answers, were recorded. We did not store personally identifiable information, collect detailed demographic or geographic attributes, or collect sensitive personal information. Human-study results are reported only in aggregate, and individual response files are not publicly released.

Risks and Intended Use The study uses simulated household activity schedules rather than real personal behavior traces, and participants were not asked to disclose personal experiences, private routines, demographic attributes, or sensitive information. We therefore consider the study to pose minimal risk to participants.

Dataset Collection Our dataset and benchmark are constructed from recorded gameplay in *The Sims 4*, used as a controllable simulation environment for non-commercial academic research. Data collection is controlled through custom scripts and community-developed modifications, with recordings captured from the perspective of an in-game robot agent and organized into video observations, event sequences, behavioral annotations, and learning targets. The original game narrative, artistic presentation, character design, and gameplay experience are not the object of study, and no models are trained on the collected data. Data collection is limited to simulated daily activity; the *Discover University* downloadable content and community-developed modifications are used only to support and control in-game robot-agent behavior.

REFERENCES

Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. Agentic AI: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. doi: 10.1109/ACCESS.2025.3532853.

Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pp. 3–14, USA, 1995. IEEE Computer Society. ISBN 0818669101.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664. Association for Computational Linguistics, July 2025b. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.183. URL https://aclanthology.org/2025.acl-long.183/.

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. *arXiv preprint arXiv:2601.03515*, 2026.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

M. M. Bongard. *Pattern Recognition*. Spartan Books, New York, 1970. URL https://archive.org/details/patternrecogniti00bong. Translated from Russian.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Guo Chen, Lidong Lu, Yicheng Liu, Liangrui Dong, Lidong Zou, Jixin Lv, Zhenquan Li, Xinyi Mao, Baoqi Pei, Shihao Wang, Zhiqi Li, Karan Sapra, Fuxiao Liu, Yin-Dong Zheng, Yifei Huang, Limin Wang, Zhiding Yu, Andrew Tao, Guilin Liu, and Tong Lu. Towards multimodal lifelong understanding: A dataset and agentic baseline, 2026. URL https://arxiv.org/abs/2603.05484.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A New Challenge for Frontier AI Reasoning Systems. *arXiv preprint arXiv:2505.11831*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. RoboMME: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

Junze Deng, Qinhang Wu, Peizhong Ju, Sen Lin, Yingbin Liang, and Ness Shroff. Unlocking the power of rehearsal in continual learning: A theoretical perspective. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=p6nhzZ9ilZ.

H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16:264–285, 1969. URL https://api.semanticscholar.org/CorpusID:1177942.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. AutoGuide: Automated generation and selection of context-aware guidelines for large language model agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mRIQz8Zd6O.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=CTr3bovS5F. Survey Certification.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate

Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Google. Gemini 3 flash preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview, December 2025. Model ID: `gemini-3-flash-preview`. Accessed: 2026-05-20.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18973–18990, 2022. doi: 10.1109/CVPR52688.2022.01842.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. InductionBench: LLMs fail in the simplest complexity class. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26526–26546, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1287. URL https://aclanthology.org/2025.acl-long.1287/.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21502–21521. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jelassi24a.html.

John R Josephson and Susan G Josephson. *Abductive inference: Computation, philosophy, technology*. Cambridge University Press, 1996.

Greg Kamradt. Needle In A Haystack: Pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023. GitHub repository. Accessed: 2026-05-25.

David Klahr and Kevin Dunbar. Dual space search during scientific reasoning. *Cognitive science*, 12 (1):1–48, 1988.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*, 2024.

Chunyang Li, Weiqi Wang, Tianshi Zheng, and Yangqiu Song. Patterns over principles: The fragility of inductive reasoning in LLMs under noisy observations. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19608–19626, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1006. URL https://aclanthology.org/2025.findings-acl.1006/.

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. MIRAGE: Evaluating and explaining inductive reasoning process in language models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=tZCqSVncRf.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PMz29A7Muq.

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958. doi: 10.1147/rd.22.0159.

Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. Towards a holistic landscape of situated theory of mind in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1011–1031, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.72. URL https://aclanthology.org/2023.findings-emnlp.72/.

Ziqiao Ma, Xueyang Yu, Haoyu Zhen, Yuncong Yang, Joyce Chai, and Chuang Gan. Fast spatial memory with elastic test-time training. *arXiv preprint arXiv:2604.07350*, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=JVlWseddak.

Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, 1997. doi: 10.1023/A:1009748302351. URL https://doi.org/10.1023/A:1009748302351.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth . In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 0215, Los Alamitos, CA, USA, April 2001. IEEE Computer Society. doi: 10.1109/ICDE.2001.914830. URL https://doi.ieeecomputersociety.org/10.1109/ICDE.2001.914830.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Run Peng, Ziqiao Ma, Amy Pang, Sikai Li, Zhang Xi-Jia, Yingzhuo Yu, Cristian-Paul Bara, and Joyce Chai. Communication and Verification in LLM Agents towards Collaboration under Information Asymmetry, 2025. URL https://arxiv.org/abs/2510.25595.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. VirtualHome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4znwzG92CE.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Machel Reid, Nikolay Savinov, Dmitri Teplyashin, Kirthevasan Audhkhasi, Jean-Baptiste Alayrac, A. Cassirer, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Jiawei Ren, Yan Zhuang, Xiaokang Ye, Lingjun Mao, Xuhong He, Jianzhi Shen, Mrinaal Dogra, Yiming Liang, Ruixuan Zhang, Tianai Yue, Yiqing Yang, Eric Liu, Ryan Wu, Kevin Benavente, Rajiv Mandya Nagaraju, Muhammad Faayez, Xiyan Zhang, Dhruv Vivek Sharma, Xianrui Zhong, Ziqiao Ma, Tianmin Shu, Zhiting Hu, and Lianhui Qin. SimWorld: An Open-ended Realistic Simulator for Autonomous Agents in Physical and Social Worlds, 2026. URL https://arxiv.org/abs/2512.01078.

Gerard Salton, James Allan, and Chris Buckley. Automatic structuring and retrieval of large text files. *Commun. ACM*, 37(2):97–108, February 1994. ISSN 0001-0782. doi: 10.1145/175235.175243. URL https://doi.org/10.1145/175235.175243.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. URL https://arxiv.org/abs/1912.01734.

Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin (eds.), *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996, Proceedings*, Lecture Notes in Computer Science, pp. 3–17. Springer, 1996. doi: 10.1007/BFB0014140. URL https://doi.org/10.1007/BFb0014140.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283, June 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025. URL https://arxiv.org/abs/2509.17765.

Jiaqi Yan, Ruilong Ren, Jingren Liu, Shuning Xu, Ling Wang, Yiheng Wang, Xinlin Zhong, Yun Wang, Long Zhang, Xiangyu Chen, Changzhi Sun, et al. TeleEgo: Benchmarking egocentric ai assistants in the wild. *arXiv preprint arXiv:2510.23981*, 2025a.

Kai Yan, Zhan Ling, Kang Liu, Yifan Yang, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. MIR-Bench: Benchmarking LLM's Long-Context Intelligence via Many-Shot In-Context Inductive Reasoning. *arXiv preprint arXiv:2502.09933*, 2025b.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, Bo Li, and Ziwei Liu. Egolife: Towards egocentric life assistant, 2025. URL https://arxiv.org/abs/2503.03803.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=G2Q2Mh3avow.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025.

Qinhong Zhou, Hongxin Zhang, Xiangye Lin, Zheyuan Zhang, Yutian Chen, Wenjun Liu, Zunzhe Zhang, Sunli Chen, Lixing Fang, Qiushi Lyu, Xinyu Sun, Jincheng Yang, Zeyuan Wang, Bao Chi Dang, Zhehuan Chen, Daksha Ladia, Quang Vinh Dang, Jiageng Liu, and Chuang Gan. Virtual community: An open world for humans, robots, and society. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Qo0OZZoTLh.

APPENDIX

A Limitations **19**

B Controlled Generation of Behavioral Phenomena **19**

B.1 Actions . . . 19
B.2 Conditions . . . 20
B.2.1 Temporal Conditions . . . 20
B.2.2 Activity-Based Conditions . . . 21
B.2.3 Condition Merging . . . 21
B.3 Profile . . . 21
B.4 Phenomenon . . . 22
B.4.1 Evolution Rules . . . 22
B.4.2 Phenomena Generation with LLMs . . . 23
B.5 Schedule Generation . . . 23
B.5.1 Activity Specification . . . 23
B.6 Video Chain . . . 24
B.7 Illustrative End-to-End Example . . . 25

C Task Design **25**

C.1 Setup: the Day Condition Set . . . 26
C.2 The Four Question Types . . . 26
C.3 Hint Levels . . . 27
C.4 History Questions . . . 28
C.5 Question Anatomy . . . 28
C.6 Selecting a Noise Condition . . . 29
C.7 Generation and Rendering . . . 30

D Hypothesis Elimination For Task Insertion **30**

D.1 Hypothesis space . . . 30
D.2 Elimination and label uniqueness . . . 31
D.3 Example: weekday-based activity . . . 31

E Dialogue Generation and Adaptation **32**

E.1 Original Dialogue Generation . . . 32
E.2 Prompted Dialogue Adaptation . . . 33
E.3 Constraints on Adapted Dialogue . . . 33
E.4 Quality Control . . . 34

F SIMLIFE Web Interface **34**

F.1 Simulated Day Timeline 34
F.2 Phenomenon Generation 34
F.3 Task Annotation 36

G Human Study **37**
G.1 User Interface 38
G.2 Anchor-Based Participant Calibration 38
G.3 Results 39

H Additional Results **39**

I Case Study Analysis of Model Reasoning Traces **42**
I.1 Hint Levels and Input Modalities 42
I.2 Trajectory Probing 42

J Vision-based Baseline Implementation Details **44**
J.1 Temporal Window Definition 44
J.2 Phase 1: Per-Window Diary Generation 44
J.3 Phase 2: Text-Only QA over Diary Summaries 45
J.4 Multi-Day Chain Handling 46
J.5 Agentic Baselines 46

K Text-Only Baseline Implementation Details **47**
K.1 Input Representation 47
K.2 Prompt Structure 47
K.3 Retry Policy 48

A LIMITATIONS

While **SIMLIFE** and **SIMLIFE-BP** have comprehensively revealed models' pattern understanding ability, several limitations remain. First, the behavioral patterns in **SIMLIFE-BP** are deterministic abstractions of real human behavior. Real-world habits are often stochastic, multi-factorial, and shaped by latent causes that may not be directly observable. We adopt deterministic rules not to claim that human behavior is rule-based, but to provide a controlled and verifiable setting in which pattern understanding can be evaluated with reliable ground truth. We view this as an intermediate step toward more realistic long-term human modeling, where the same core abilities: discovering regularities, ruling out inconsistent explanations, and adapting to change remain necessary.

At the same time, although **SIMLIFE** provides rich multimodal observations and controllable long-term behavior, simulated household routines cannot fully capture the variability, ambiguity, and social complexity of real human lives. An important next step is to study whether models trained or evaluated on **SIMLIFE-BP** transfer to real-world long-term observation datasets, and whether simulated data can serve as useful supervision for building more adaptive assistants.

Additionally, our human study is not a perfectly symmetric comparison between humans and models. To reduce cognitive load, human participants are shown a calendar-like interface and simplified structured summaries rather than raw ultra-long logs or videos (see Appendix G for more details). In dynamic-pattern settings, humans are also explicitly informed when a pattern may change, whereas models receive the corresponding signal only through the observation stream. These design choices make the human study more interpretable and feasible, but they also mean that human-model comparisons should be read as diagnostic rather than strictly controlled head-to-head evaluations.

B CONTROLLED GENERATION OF BEHAVIORAL PHENOMENA

SIMLIFE inverts the usual record-then-annotate pipeline into a controlled generation procedure: we specify a latent behavioral rule first, then assemble a long video that provably exhibits it by selecting and linking recordings from a finite pool of reusable pre-recorded days. This allows efficient benchmark construction with minimal annotation and filtering effort, and guarantees some extent of reproducibility. [3]

The abstractions below build around one central object, the **Phenomenon**: a complete specification of a behavioral rule, and the blueprint every generated episode follows. Below it, an action, a condition, and a profile are the successive units that compose a phenomenon out of smaller parts. Above it, a schedule and a video chain are the successive steps that turn a phenomenon into an actual sequence of recordings. We walk through these six layers in that order, then trace one complete example end to end in B.7.

B.1 ACTIONS

An **action** is the atomic unit of behavior in **SIMLIFE**. Each action belongs to one of ten categories and can be performed by one or more characters ("Sims"), each in a designated **role** (e.g., `cook` or `doer`) that the matcher in B.5.1 selects on.

Characters. The household consists of five characters: *Father Sim*, *Mother Sim*, *Son Sim*, *Daughter Sim*, and *Servo Bot* (a household robot). All the videos are recorded from the egocentric view of *Servo Bot*.

Action categories. Table 4 summarizes the ten action categories. Cooking recipes additionally carry dietary tags (*lactose-free*, *vegetarian*) that enable dietary-aware condition design.

Time periods. A simulated day spans from 6 AM to 10 PM.

[3] Reproducibility is bounded by the simulator. A deterministic environment (e.g. with a controllable random seed) gives full reproducibility; *The Sims 4* does not — we can ensure Bob drinks coffee around 8 am but not the exact time, nor fine-grained details such as which hand holds the cup.

Category	Example Actions	Count	Collab.
Cooking	{eggs_and_toast, spaghetti, butter_chicken}	33	✓
Piano	{practice_piano, play_blues_songs_on_piano}	5	–
Guitar	{practice_guitar, play_rock_songs_on_guitar}	5	–
Violin	{practice_violin, play_country_songs_on_violin}	5	–
Fitness	{endurance_run, box, heavy_lifting, jogging}	5	–
Fishing	{fish, fish_with_bait}	2	✓
Writing	{practice_writing, write_fantasy_book, write_biography}	8	–
Video Games	{play_sims_forever, play_blicblock}	5	–
Logic	{play_chess}	1	✓
General	{brush_teeth, drink_coffee, homework, watch_tv}	10	Mixed

Table 4: Action categories in **SIMLIFE**. "Collab." indicates whether the action can involve multiple participants.

B.2 CONDITIONS

A **condition** turns an abstract behavioral rule into per-day ground truth: it determines *on which days* a target activity should or should not occur, and the resulting per-day labels are exactly what the benchmark later queries. Each condition targets an *activity*, an action performed within a time window by a specified set of participants, which we denote $\mathcal{A}$ and formalize fully, as an **Activity Specification**, once it is needed for matching against recordings (B.5). Given a calendar of N days, a condition produces a binary label vector $\mathbf{l} \in \{0,1\}^N$, where $l_d = 1$ indicates that the activity is required on day d, and $l_d = 0$ means the activity is required not to happen on day d.

There are two families of conditions, corresponding to the two ways a rule can be driven: *temporal* conditions are driven by calendar structure (e.g., the day of the week), and *activity-based* conditions are driven by another activity (e.g., an event the day before). Because real behavioral patterns are rarely a single rule but a conjunction, exclusion, or alternation of simpler ones, conditions targeting the same activity can be merged; each condition therefore carries an `operation` field governing how it combines with others (B.2.3).

B.2.1 TEMPORAL CONDITIONS

Temporal conditions determine activity occurrence based purely on time, frequency, or calendar structure (see Table 5). These conditions can be used to create structured daily life patterns that depend on the time of day.

Type	Template	Example
`every_selected_weekday`	On {list of weekdays}	"Practice piano on Monday, Wednesday, Friday"
`repeat_every_other`	Repeat {count} times every other {value} {unit}	"Go fishing 1 time every other week"
`on_date`	On {date_expression}	"Write screenplay on the last Friday of each month"

Table 5: Temporal condition types. Parameters in braces are configurable.

Duration units. The {unit} parameter supports six values, each defining a window length for grouping calendar dates as is shown in Table 6.

Unit	Length (days)	Scope
`day`	1	Every day
`week`	7	Calendar weeks
`month`	30	Calendar months
`year`	365	Calendar years
`weekday`	5	Monday–Friday only
`weekend`	2	Saturday–Sunday only

Table 6: Supported duration units and their effective lengths.

Date expressions. The `on_date` type supports patterns such as:

- `the {n}-th day of each month` (e.g., the 15th)
- `the {n}-th {weekday} of each month` (e.g., the 2nd Tuesday)
- `the last day of each month`
- `the last {weekday} of each month` (e.g., the last Friday)

B.2.2 Activity-Based Conditions

Activity-based conditions make the occurrence of a *main activity* $\mathcal{A}_{\text{main}}$ depend on a *trigger activity* $\mathcal{A}_{\text{trigger}}$ (see Table 7). The trigger activity is pre-sampled independently across all calendar dates with a configurable frequency $f \in (0, 1]$ (default $f = 0.1$). These conditions specify conditional behavioral patterns with activities that only occur when another event does.

Type	Template	Example
`right_after`	$\mathcal{A}_m$ the day after $\mathcal{A}_t$	"Drink coffee the day after a late-evening run"
`within_same_day`	$\mathcal{A}_m$ same day as $\mathcal{A}_t$	"Shower on the same day as heavy lifting"
`no_activity_right_after`	$\mathcal{A}_m$ must *not* occur the day after $\mathcal{A}_t$	"No video games the day after failing homework"
`no_activity_within_same_day`	$\mathcal{A}_m$ must *not* occur same day as $\mathcal{A}_t$	"No coffee on jogging days"

Table 7: Activity-based condition types. $\mathcal{A}_m$ = main activity, $\mathcal{A}_t$ = trigger activity.

B.2.3 Condition Merging

When multiple conditions target the same activity, their binary labels are merged day-by-day using the `operation` field as is shown in Table 8. Conditions are applied sequentially; later conditions in the list take higher priority.

Operation	Merge Rule	Use Case
`or`	$m_d \leftarrow m_d \vee l_d^{(i)}$	Additive: activity *should* happen
`and`	$m_d \leftarrow m_d \wedge l_d^{(i)}$	Restrictive: activity should *not* happen

Table 8: Condition operations and their semantics. $l_d^{(i)}$ is the label from condition i for day d; m_d is the running merged label.

Running example. Consider the phenomenon "Father Sim drinks coffee every weekday morning, but not on jogging days." This is implemented with two conditions on the same $\mathcal{A}_{\text{coffee}}$:

1. `every_selected_weekday`: on Monday, Tuesday, Wednesday, Thursday, Friday, `operation=or` $\Rightarrow$ labels coffee on each weekday.
2. `no_activity_within_same_day`: trigger = $\mathcal{A}_{\text{jogging}}$, frequency = 0.1, `operation=and` $\Rightarrow$ samples jogging events with 10% frequency, and then removes coffee on days when jogging occurs.

The merged schedule reflects the conjunction: coffee appears on weekdays *except* those with a jog.

B.3 Profile

A **Human Profile** bundles the conditions that define one stable behavioral regime for a character: a static pattern is a single profile, and a dynamic phenomenon is several profiles linked together (B.4). Formally, it is a named behavioral state observed over a specified duration:

$$H = (\textit{desc},\ N,\ T) \tag{1}$$

where N is the observation duration in days and T is a list of conditions, each either temporal or activity-based (B.2.1 and B.2.2).

While the underlying system allows a profile's conditions to target different activities (i.e., a single profile could encode several independent activity patterns at once), SIMLIFE-BP restricts every condition within a profile to share the same target activity. This keeps each task focused on inferring the rule governing one activity, rather than requiring a model to first disentangle which evidence belongs to which of several interleaved patterns.

Example: "Father Sim coffee drinker" (*length* $= 21$ days), all conditions targeting $\mathcal{A}_{\text{coffee}}$.

- `every_selected_weekday`(weekdays={Monday, Tuesday, Wednesday, Thursday, Friday}, op=`or`)
- `right_after`(trigger=$\mathcal{A}_{\text{late_run}}$, op=`or`)

This profile labels coffee on every weekday *and* on any day following a late-evening run.

B.4 PHENOMENON

A **Phenomenon** is the top-level behavioral narrative that profiles compose into and the object each generated episode ultimately corresponds to:

$$P = (\mathbf{H},\ \mathbf{E},\ \textbf{start date}) \tag{2}$$

where $\mathbf{H} = [H_1, \ldots, H_k]$ is an ordered list of profiles and $\mathbf{E} = [E_1, \ldots, E_{k-1}]$ is a list of evolution rules connecting consecutive profiles. The **start date** indicates the calendar date that the phenomenon starts to simulate. This structure enables us to assemble video chain instances for our dataset.

We support two phenomenon types (Table 9):

Type	$\lvert\mathbf{H}\rvert$	$\lvert\mathbf{E}\rvert$	Description
Pattern	1	0	Static behavioral pattern; no change over time
Dynamic	≥ 2	≥ 1	Two or more behavioral phases connected by (a) life event(s)

Table 9: Phenomenon types and their structural constraints.

B.4.1 EVOLUTION RULES

An **Evolution Rule** is needed only when a phenomenon is dynamic: it describes a transition between two profiles, triggered by a life event:

$$E = (\textit{desc},\ H_{\text{from}},\ H_{\text{to}},\ \mathcal{A}_{\text{must_have}},\ \mathcal{A}_{\text{must_not_have}}) \tag{3}$$

The transition occupies exactly one day, during which the activities in $\mathcal{A}_{\text{must_have}}$ are required and those in $\mathcal{A}_{\text{must_not_have}}$ are forbidden.

Example phenomenon for Dynamic Understanding.

> "Father Sim drinks coffee every weekday morning. After a health scare, he stops drinking coffee and switches to morning jogging."

- H_1 = "coffee drinker" (21 days): `every_selected_weekday`(M,T,W,Th,F) on $\mathcal{A}_{\text{coffee}}$
- E_1 = "health scare": $H_1 \rightarrow H_2$
- H_2 = "healthy lifestyle" (14 days): `no_activity_within_same_day`(trigger=$\mathcal{A}_{\text{jogging}}$, freq=1) on $\mathcal{A}_{\text{coffee}}$

The resulting observation spans $21 + 1 + 14 = 36$ days.

B.4.2 Phenomena Generation with LLMs

Phenomena are generated using Gemini 2.5 Flash with structured prompts that provide:

1. The full action catalog (Table 4).
2. The activity-matching schema used to express conditions, formalized as an Activity Specification in B.5.
3. All condition trigger templates (Tables 5 and 7) with examples.
4. The output JSON schema specifying profiles, conditions, and evolution rules.
5. Constraints: minimum conditions per profile (≥ 2), number of profiles, and number of evolution rules.

For *pattern* phenomena, the prompt requests one profile with no evolution rules. For *dynamic* phenomena, the prompt requests exactly two profiles and one evolution rule connecting them. Generated JSON is validated against the schema before manual verification.

While we use LLM generation for some ideas for new phenomena to add to our dataset, most of the data was created through manual annotation. A portion of the phenomena for our data was also generated automatically using simulated day activity distributions to ensure we had enough simulated day videos to create the video chain instances.

With a phenomenon specified, the remaining two sections describe how it becomes data: a schedule compiles it into a concrete, date-by-date requirement matrix (B.5), and a video chain instantiates that matrix as an ordered sequence of recordings (B.6).

B.5 Schedule Generation

A **schedule** compiles a declarative phenomenon into a concrete, date-by-date requirement: it turns "this rule holds" into "here is what day 12 must contain." The same phenomenon compiled from a different start date, or with different independently-sampled trigger days, yields a different schedule. This makes it possible to generate multiple valid schedules with one phenomenon, as well as doing rejection sampling until getting a valid schedule. Each requirement in a schedule is expressed as an **Activity Specification**, which filters the pool of recorded days and returns those that contain a matching activity.

B.5.1 Activity Specification

An **Activity Specification** (ActivitySpec) selects a class of concrete activities along three orthogonal dimensions:

$$\mathcal{A} = (\textit{actions},\ \textit{time_zones},\ \textit{allowed_participant_sets}) \tag{4}$$

We detail each dimension in Table 10. A concrete activity matches $\mathcal{A}$ if and only if *all three* dimensions are satisfied simultaneously; any recorded day containing such an activity is a candidate to satisfy this requirement.

Component	Description	Matching
`actions`	List of action names (e.g., `drink_coffee`, `jogging`)	Any one matches (OR)
`time_zones`	List of `TimeZone` (TZ) objects, each with `start_time`, `end_time`, and `type` $\in$ {`must_have`, `must_not_have`}	`must_have`: OR across zones; `must_not_have`: AND (avoid all)
`allowed_participant_sets`	List of participant combinations, each a list of (character, role) pairs	Any one set matches (OR)

Table 10: Components of an Activity Specification and their matching semantics.

Day-level scope. An ActivitySpec matches within a single recorded day and carries no information about which day it is or where it falls on the calendar; its `time_zones` are clock windows (e.g.,

06:00–10:00), not weekdays or dates. Whether a requirement should apply on a given day, such as only on weekdays, is determined separately by schedules which are defined with conditions (B.2.1).

Example 1: Father Sim drinks coffee in the morning.

$$\begin{aligned}\mathcal{A}_{\text{coffee}} = \Big(&\big[\texttt{drink_coffee}\big],\\ &\big[\text{TZ}(06{:}00,\ 10{:}00,\ \texttt{must_have})\big],\\ &\big[\big[(\text{Father Sim},\ \texttt{doer1})\big]\big]\Big)\end{aligned} \tag{5}$$

Example 2: Mother Sim or Daughter Sim does any fitness activity in the evening.

$$\begin{aligned}\mathcal{A}_{\text{fitness}} = \Big(&\big[\texttt{endurance_run},\ \texttt{jogging},\ \texttt{box}\big],\\ &\big[\text{TZ}(15{:}00,\ 22{:}00,\ \texttt{must_have})\big],\\ &\big[\big[(\text{Mother Sim},\ \texttt{doer1})\big],\\ &\ \big[(\text{Daughter Sim},\ \texttt{doer1})\big]\big]\Big)\end{aligned} \tag{6}$$

This specification matches any of the three fitness actions, performed by either Mother Sim or Daughter Sim (but not both simultaneously), between 15:00 and 22:00.

Given a phenomenon P with profiles $\mathbf{H}$ and a start date, the schedule generator produces a mapping from calendar dates to activity requirements:

$$\mathcal{S} : \textit{date} \to \{(\mathcal{A}_j,\ b_j)\}_{j=1}^{m} \tag{7}$$

where m is the number of distinct ActivitySpecs referenced by the phenomenon, one for each profile's target activity plus one for each trigger activity used by an activity-based condition, and $b_j \in \{0, 1\}$ indicates whether activity specification $\mathcal{A}_j$ is required on that date. We use these schedules to select simulated day videos and assemble a unique, long video for each phenomenon that shows enough evidence of the pattern the phenomenon describes.

Algorithm. For each profile H_i:

1. **Pre-sample trigger activities.** For each unique trigger activity spec across all activity-based conditions, independently sample each date as active with probability f (B.2.2).
2. **Generate per-condition labels.** Dispatch each condition to its handler (Tables 5–7), producing binary labels.
3. **Merge labels.** Apply each condition's `operation` (Table 8) sequentially to produce the final schedule.

Constraint matrix. The schedule is converted into a boolean matrix $\mathbf{M} \in \{0, 1\}^{N \times m}$, where $M_{d,j} = 1$ if day d requires activity spec $\mathcal{A}_j$. Each row defines a unique constraint pattern that Video Chain construction (B.6) matches against the pool of pre-recorded simulated days.

B.6 Video Chain

A **Video Chain** instantiates a schedule: it realizes the abstract requirement matrix $\mathbf{M}$ as an ordered sequence of actual recordings, and is the final data instance inserted into the dataset.

$$\mathcal{V} = [s_1, s_2, \ldots, s_N] \tag{8}$$

where each s_d is a pre-recorded simulated day whose activities satisfy the constraint pattern at position d.

Construction. For each unique constraint pattern (row of $\mathbf{M}$), the system queries the database for simulated days whose recorded activities match all required and forbidden activity specifications (B.5.1), filtered by weekday/weekend status. Days are sampled without replacement; if the pool is exhausted, it resets for round-robin reuse.

Multi-profile chains. For dynamic phenomena, the video chain is built segment by segment:

$$\mathcal{V} = [\underbrace{s_1, \ldots, s_{|H_1|}}_{\text{Profile } H_1}, \underbrace{s_{|H_1|+1}}_{\text{Evolution}}, \underbrace{s_{|H_1|+2}, \ldots, s_N}_{\text{Profile } H_2}] \tag{9}$$

The evolution rule day must satisfy $\mathcal{A}_{\text{must_have}}$ and avoid $\mathcal{A}_{\text{must_not_have}}$.

Each day in the chain carries two metadata records:

- **Day Activity Occurrence Set**: which activity specs were required/forbidden.
- **Day Condition Set**: which conditions were satisfied (boolean flags per condition). This is used for generating counterfactual questions.

B.7 Illustrative End-to-End Example

We trace the pipeline described above, end to end, for one pattern phenomenon:

> Son Sim practices guitar every weekday evening. On days when Daughter Sim plays violin, Son Sim also plays guitar within the same day.

Step 1: Activity Specifications.

$$\begin{aligned}\mathcal{A}_{\text{guitar}} = \big(&[\texttt{practice_guitar}],\\ &[\text{TZ}(15{:}00, 22{:}00, \texttt{must_have})],\\ &[[(\text{Son Sim}, \texttt{doer1})]]\big)\\ \mathcal{A}_{\text{violin}} = \big(&[\texttt{practice_violin}],\\ &[\text{TZ}(15{:}00, 22{:}00, \texttt{must_have})],\\ &[[(\text{Daughter Sim}, \texttt{doer1})]]\big)\end{aligned}$$

Step 2: Conditions.

1. C_1 = `every_selected_weekday`(weekdays= {Monday, Tuesday, Wednesday, Thursday, Friday}, op=`or`) on $\mathcal{A}_{\text{guitar}}$: guitar every weekday.
2. C_2 = `within_same_day`(trigger=$\mathcal{A}_{\text{violin}}$, freq=0.1, op=`or`) on $\mathcal{A}_{\text{guitar}}$: guitar on violin days too.

Step 3: Profile.

$$\mathcal{H} = (\text{“Son guitar practice”},\ 30,\ [C_1, C_2]).$$

Step 4: Schedule. Over 30 days, the merged schedule labels guitar on all weekdays (from C_1) plus any weekend day when violin is triggered (from C_2, with 10% probability per day).

Step 5: Video chain. The constraint matrix distinguishes three day types: (a) guitar required, (b) both guitar and violin required, and (c) neither required. For each type, matching simulated days are sampled from the database and assembled into a 30-day video chain.

C Task Design

Each **SimLife-BP** task poses four types of multiple-choice questions over the same long video: *direct prediction*, *counterfactual prediction*, *noisy counterfactual prediction*, and *inverse counterfactual reasoning*. All four are built on a single abstraction, the *Day Condition Set* (Appendix B.6), and together probe whether a model has recovered the rule governing a character's behavior.

The design factors into a shared foundation and three orthogonal axes. The foundation is the *Day Condition Set* (Appendix C.1): the truth table of the behavioral rule a task targets. The three axes are *what* is asked (the four question types, Appendix C.2), *how much* of the rule is revealed (hint levels, Appendix C.3), and *which* behavioral phase is targeted (standard vs. history questions, Appendix C.4). Appendix C.5 composes the three axes into one rendered question, and Appendices C.6 and C.7 document two implementation details: noise-condition selection and the natural-language rendering pass.

C.1 Setup: the Day Condition Set

For every day in a video chain we record two things: (i) the boolean flag for each condition that governs the *main activity* (e.g., did "today is Saturday" hold? or did "Daughter Sim played violin the day before" hold?), and (ii) the **outcome** $M \in \{0, 1\}$, whether the main activity actually occurred that day. We call this record the **Day Condition Set** (DCS): a condition configuration together with its outcome. We write $M(d)$ for the outcome of a DCS d.

Across a chain only a few distinct DCSs appear; collecting them yields the *truth table* of the phenomenon's underlying rule. A task is anchored to one **observed** DCS d_{obs}. The four question types are different ways of interrogating this truth table.

Running example We use one phenomenon throughout this section:

> Son Sim plays chess in the evening (3PM–10PM) on weekends, and also on any day right after Daughter Sim plays violin.

The main activity $\mathcal{A}_{\text{chess}}$ is governed by two conditions, merged with `or`:

- C_1 (temporal): *today is Saturday or Sunday*.
- C_2 (activity, `right_after`): *Daughter Sim played violin the day before*.

Across the chain the days fall into four distinct DCSs, giving the truth table in Table 11.

DCS	C_1 (weekend)	C_2 (violin yesterday)	M (chess occurs)
d_0	F	F	0
d_1	T	F	1
d_2	F	T	1
d_3	T	T	1

Table 11: Truth table for the chess phenomenon ($M = C_1 \vee C_2$).

C.2 The Four Question Types

The four types are different interrogations of the same truth table (Table 12). We illustrate each on the running example, taking the observed DCS to be $d_{\text{obs}} = d_2$: a weekday following a violin day, on which chess *did* occur ($M(d_2) = 1$).

Type	Direction	Answer	What it tests
Direct	—	$M(d_{\text{obs}})$	reading the actual outcome
Counterfactual	cause → effect	$M(d_{\text{cf}})$	causal propagation
Noise counterfactual	cause → effect	$M(d_{\text{obs}})$ (unchanged)	resistance to spurious cues
Inverse counterfactual	effect → cause	condition matching $M(d_{\text{inv}})$	abductive (backward) inference

Table 12: The four question types. d_{obs} is the observed DCS; d_{cf} and d_{inv} are alternative DCSs chosen for the counterfactual and inverse types.

Direct Prediction The baseline. It asks about the observed day exactly as recorded, and the answer is the true outcome $M(d_{\text{obs}})$; no hypothetical reasoning is required. This measures whether the model

can read off what actually happened, and serves as the reference point against which the counterfactual variants are scored. *Example: "Will Son Sim play chess this evening?"* → **Yes** ($M(d_{\text{obs}}) = 1$).

Counterfactual Prediction We select a *different* DCS d_{cf} from the same truth table and ask what would happen under it; the answer is $M(d_{\text{cf}})$. This requires models to situate themselves into the counterfactual condition and predict the occurrence of the target activity accordingly. *Example:* Let $d_{\text{cf}} = d_0$ ($M(d_0) = 0$). *"If Daughter Sim had not played violin the day before, would Son Sim play chess this evening?"* → **No**. Note that we only require $d_{\text{obs}} \neq d_{\text{cf}}$, thus sometimes the outcome of direct prediction would be the same as counterfactual prediction.

Noise Counterfactual Prediction Structurally a counterfactual, but the introduced condition is *causally irrelevant* to the main activity, such as an unrelated action by another character. Because the noisy condition does not enter the truth table, the answer stays at $M(d_{\text{obs}})$. This requires models to reason that the given condition is independent to the occurrence of the target activity, thus answer the question based on the current observation. *Example: "If Father Sim had washed the dishes yesterday, would Son Sim play chess this evening?"* → **Yes**, unchanged ($M(d_{\text{obs}}) = 1$).

Inverse Counterfactual Reasoning We assert the negated outcome as a premise (premise $= \neg M(d_{\text{obs}})$) and ask which *conditions* are consistent with it, choosing between a condition statement drawn from an alternative DCS d_{inv} and its opposite. The answer follows from matching the premised outcome against $M(d_{\text{inv}})$. Unlike the forward types, this requires models to reason abductively, inferring a plausible cause from a stated effect rather than propagating a known cause to its effect. *Example:* $d_{\text{inv}} = d_0$ ($M(d_0) = 0$) *"If Son Sim had not played chess this evening, which statement is more likely to be true? (A) Daughter Sim did not play violin the day before; (B) Daughter Sim played violin the day before."* → **(A)**. Under the `or` rule on a non-weekend day, chess can fail only if the violin trigger was also absent, so the model reasons from effect back to cause.

Each type is rendered in two surface formats: a **yes/no** form ("Will X ... today?") and a **two-option** form ("which statement is more likely to be true?"), whose templates are collected in Table 13.

Type	Format	Body template
Direct	yes/no	`Will {sim} {action} today{time}?`
	two-option	`Which statement is more likely to be true?`
Counterfactual	yes/no	`If {conditions of d_cf}, will {sim} {action} today{time}?`
	two-option	`If {conditions of d_cf}, which statement is more likely to be true?`
Noise counterfactual	yes/no	`If {noise}, will {sim} {action} today{time}?`
	two-option	`If {noise}, which statement is more likely to be true?`
Inverse counterfactual	yes/no	`If {M}, is the statement "{conditions of d_inv}" likely to be true?`
	two-option	`If {premise}, which statement is more likely to be true?`

Table 13: Question Body templates by type and format. `{time}` is an optional clock range (e.g. *between 3PM and 10PM*). *Yes/no* options are "`Yes`"/"`No`"; *two-option* choices are "`{sim} will (not) {action} today{time}`" for the first three types, and the two condition statements `{icf_body}` vs. `{icf_body_opposite}` for inverse counterfactual. Templates are emitted in the indicative; the LLM rendering pass (Appendix C.7) keeps direct questions indicative and converts the three counterfactual types to the subjunctive.

C.3 Hint Levels

The four question types fix *what* is asked; the **hint level** fixes *how much of the governing rule is revealed* before the model must answer. We expose three hint levels as detailed in Table 14.

Each governing condition contributes one clause, rendered differently at the partial and full levels (Table 15). Here `{participant}` is a character (e.g. *Daughter Sim*), `{activity}` the natural-language action (e.g. *playing violin*), and `{window}` an optional clock range "`from HH AM/PM to HH AM/PM`" omitted when the activity spans the default observable day (6AM–10PM).

Level	Rendered as	What the model is told
No hint	`<body>`	nothing about the rule
Partial hint	`<body>` (`<hint$_p$>`)	condition categories + logic
Full hint	`<body>` (`<hint$_f$>`)	exact condition bindings

Table 14: The three hint levels. `<body>` refers to the four question types introduced in Appendix C.2.

Family	Condition Type	Partial Hint	Full Hint
Temporal	`every_selected_weekday`	the day of the week	weekday names, e.g. `Monday, Wednesday`; all seven → `from Sunday to Saturday`
Temporal	`on_date` (date form)	the date in the month	the `{ordinal}` of each month; `the last day of each month`
Temporal	`on_date` (weekday form)	a specific weekday of each month	the `{ordinal} {Weekday}` of each month
Temporal	`repeat_every_other`	related to whether a specific action is done by `{participant}{window}` the day before	related to whether `{participant}` does `{activity}{window}` the day before
Activity	`right_after`	related to a specific action done by `{participant}` the day before`{window}`	related to `{participant}` doing `{activity}{window}` the day before
Activity	`within_same_day`	related to a specific action done by `{participant}` today`{window}`	related to `{participant}` doing `{activity}{window}` within the same day
Activity	`no_activity_right_after`	related to a specific action done by `{participant}` the day before`{window}`	related to `{participant}` doing `{activity}{window}` the day before
Activity	`no_activity_within_same_day`	related to a specific action done by `{participant}` today`{window}`	related to `{participant}` doing `{activity}{window}` within the same day

Table 15: Per-condition hints at the partial and full levels.

For the running example (Appendix C.1), the three levels of the direct question read:

- **No Hint:** *"Will Son Sim play chess today from 3PM to 10PM?"*
- **Partial Hint:** *"... (It is related to the day of the week and to a specific action done by Daughter Sim the day before.)"*
- **Full Hint:** *"... (It is related to whether today is Saturday or Sunday and to whether Daughter Sim played violin the day before.)"*

C.4 History Questions

SimLife-BP also includes **history** questions (or `Hist` in 4.3.2), which ask about an *earlier* behavioral phase of a dynamic phenomenon: a period governed by the rule that held *before* a life event changed it (Appendix B.4.1). Because these questions concern a past period rather than "today," each is prepended with a temporal **anchor** that re-situates the model in the corresponding phase. We design two types of temporal anchors (Table 16) that coarsely identify the relevant historical phase, encouraging models to reason about the previous pattern rather than retrieve a specific past day. History questions inherit the three counterfactual variants from standard questions[4], but omit direct prediction, since asking about an imagined past is counterfactual by default.

C.5 Question Anatomy

Every question is issued in one of two surface formats: a **yes/no** form ("... will X ... today?") and a **two-option** form ("which statement is more likely to be true?"). The three axes above—question type, hint level, and task kind—then compose into a single template. Every rendered question, in either surface format, is the concatenation of three slots:

$$\underbrace{\langle \text{history prefix} \rangle}_{\text{history tasks only}} \quad \underbrace{\langle \text{question body} \rangle}_{\text{always}} \quad \underbrace{(\langle \text{hint} \rangle)}_{\text{partial / full only}}$$

[4]We use standard questions to refer to questions about the current pattern.

Anchor	Prefix template / example
Evolution-description	`Back before {event}, <body>` *"Back before Son Sim started a new school term, . . . "*
Vague-days	`{vague_phrase}, <body>` *"About a month earlier, . . . "*

Table 16: History anchors. `{vague_phrase}` is one phrase sampled deterministically per task (shared across its questions) from a table bucketed by how far back the phase sits, e.g. 22–28 → "About three weeks earlier," 29–35 → "About a month earlier."

The **body** is selected by question type and answer format (Table 13); the **history prefix** is supplied for history tasks (Table 16, empty otherwise); and the **hint** is appended at the partial or full level (Tables 14–15, empty at no-hint).

A fully populated example—a *history*, *counterfactual*, *partial-hint* question—exercises all three slots:

> **[prefix]** Back before Son Sim started a new school term, **[body]** if Daughter Sim had not played violin the day before, would Son Sim play chess this evening? **[hint (partial)]** (It is related to the day of the week and to a specific action done by Daughter Sim the day before.)

C.6 Selecting a Noise Condition

A noise counterfactual is only meaningful if the introduced condition is genuinely uninformative about the main activity. We do not pick such a condition by hand or by semantic intuition; instead we require it to be *empirically independent* of the main activity A within the specific video chain, and we verify this with a 2×2 contingency table.

Four-cell independence test. First, we use the occurrence of an action as the noise condition. For a candidate action X we walk every day of the chain and tally it by two booleans: whether X occurred in the relevant window (under the same temporal semantics as the ground-truth condition, e.g. `right_after` checks day $T-1$), and whether the main activity A occurred that day. This yields the table in Table 17.

	A **occurred**	A **did not**
X **occurred**	≥ 1	≥ 1
X **did not occur**	≥ 1	≥ 1

Table 17: Independence test for a candidate noise action X. All four cells must be observed at least once for X to qualify.

X qualifies only when *all four* cells are non-empty: the chain must contain days where X co-occurs with A happening, with A not happening, and likewise days without X under both outcomes. A populated table demonstrates that the presence of X carries no information about A, so X is a true distractor rather than an accidental correlate. *Edge case:* when A is invariant across the chain (it structurally always or never occurs, so only one outcome row is reachable), independence cannot be shown from a single row; the test then relaxes to requiring X itself to vary: both X occurred and X did not occur must appear in the reachable row.

Candidate pool. The candidate X is drawn from the same condition-slot machinery used by hypothesis elimination, so the noise action is checked under exactly the same lookup semantics as a real condition would be. When the ground truth contains an activity condition, we reuse the first one's slot parameters (participant, time zones, action-hierarchy level, and condition type) and search its action pool, excluding the true cause itself. When the ground truth is purely temporal, we synthesize a fixed condition `right_after`, the same participant as the main activity, a full

observable-day window (06:00–22:00), and subcategory level. Socially entangled action categories are excluded from the pool, since their occurrence is too context-dependent to serve as clean noise. The search is exhaustive over the pool and returns the first X that passes the test; if none does, no noise counterfactual is generated for that task.

C.7 Generation and Rendering

Questions are produced in two stages. First, a templating stage emits each question with structured metadata: the differing conditions for a counterfactual (`if_clause`), the irrelevant action for a noise counterfactual (`noise_framing`), and the negated outcome plus candidate condition lists for an inverse counterfactual (`premise`, `icf_body`, `icf_body_opposite`). Second, an LLM rendering pass converts these templates into fluent, mood-correct natural language: indicative for direct questions and subjunctive for the three counterfactual types, preserving all logical content and the exact character names and clock-time spans.

D Hypothesis Elimination For Task Insertion

A benchmark task is inserted at a given point only if every plausible reading of the hint that is still consistent with what has been observed so far agrees on the answer. This section formalizes that criterion: we enumerate every hypothesis a hint permits, discard those contradicted by the recorded days, and query the model only once the survivors all agree. We use the partial-hint setting as the reference, since observations, hint, and question together guarantee this criterion is satisfiable under both the partial-hint and full-hint settings. In the no-hint setting, sufficiency cannot be formally defined because the hypothesis space is unbounded; we use it instead to evaluate open-ended pattern reasoning under conditions closer to the real world.

We ground each definition below in one running example: a partial hint stating that a target activity "painting" is related to the day of the week, with the true rule (unknown to the reasoner) being "happens on Mondays and Fridays."

D.1 Hypothesis space

Condition instantiations. Consider a target activity governed by K latent conditions $C_1, \ldots, C_K$. Each condition is either temporal or activity-based. For each condition C_i, the hint level defines a finite hypothesis space Ω_i of the values that condition could take. For our running example, $K = 1$ and the hint restricts Ω_1 to the $2^7 - 1 = 127$ non-empty subsets of weekdays W.

A candidate condition tuple is

$$\omega = (\omega_1, \ldots, \omega_K) \in \Omega_1 \times \cdots \times \Omega_K.$$

On day t, this tuple induces a binary condition vector

$$\mathbf{b}(\omega, t) = (b_1(\omega_1, t), \ldots, b_K(\omega_K, t)) \in \{0, 1\}^K$$

where $b_i(\omega_i, t) = 1$ iff the corresponding condition is satisfied. In our example, $\omega = W$ and $b_1(W, t) = \mathbf{1}[t \in W]$: a single bit saying whether day t's weekday falls in the candidate subset W.

Boolean outcome rules. A hint tells the reasoner *which conditions* might matter, but not *how* they combine with the outcome. That mapping is itself unknown and must be enumerated. We represent it as a Boolean rule

$$g : \{0, 1\}^K \to \{0, 1\},$$

mapping condition configurations to activity outcomes. Enumerating over all such g, rather than assuming the hinted conditions are positively associated with the activity, is what lets this framework stay agnostic to how **SimLife** actually generates data: the benchmark's condition merging uses only `or`/`and` (B.2.3), but a reasoner without that knowledge must consider every possibility, including rules where the condition suppresses the activity, has no effect, or the activity always or never occurs. For $K = 1$, there are $2^{2^1} = 4$ such rules; applied to our example, these are "always paint," "never paint," "paint iff $t \in W$," and "paint iff $t \notin W$."

Full hypotheses. A full candidate hypothesis is

$$\theta = (\omega, g),$$

with prediction

$$\hat{o}_t(\theta) = g\big(\mathbf{b}(\omega, t)\big).$$

The full candidate set is

$$\Theta = \left\{ (\omega, g) : \begin{array}{l} \omega \in \Omega_1 \times \cdots \times \Omega_K, \\ g : \{0,1\}^K \to \{0,1\} \end{array} \right\},$$

of size

$$|\Theta| = \Big(\prod_{i=1}^{K} |\Omega_i|\Big) \cdot 2^{2^K},$$

where the first factor enumerates condition instantiations and the second enumerates Boolean outcome rules. For our running example, $|\Theta| = 127 \times 4 = 508$. More generally, since our benchmark uses at most two conditions, the Boolean-rule factor is 4 for $K = 1$ and 16 for $K = 2$.

D.2 Elimination and label uniqueness

Sequential hypothesis elimination. Hypothesis elimination proceeds cumulatively over the observed days. Let $o_t \in \{0, 1\}$ indicate whether the target activity occurs on day t. Starting from the full candidate set Θ, we eliminate any hypothesis whose prediction disagrees with the observed outcome. The remaining set after day t is

$$\mathcal{R}_t = \{\theta \in \mathcal{R}_{t-1} : \hat{o}_t(\theta) = o_t\}, \qquad \mathcal{R}_0 = \Theta.$$

Because the true generating rule produced every observation by construction, it is never contradicted and therefore never eliminated: $\mathcal{R}_t$ always contains it and is never empty. Checking that every hypothesis in $\mathcal{R}_t$ agrees is therefore equivalent to checking that they all agree with the true rule, even without knowing which surviving hypothesis that is.

Convergence day. Let $\mathcal{D}$ denote the set of Day Condition Sets (DCSs) associated with the behavioral pattern, and let $d_t \in \mathcal{D}$ denote the DCS instantiated on day t. For a sequence of T days, we define the *convergence day* Y as the earliest day satisfying both

$$\mathcal{R}_t = \mathcal{R}_Y \qquad \forall t \geq Y,$$

and

$$\forall d \in \mathcal{D}, \quad \exists t \in Y, \ldots, T \text{ such that } d_t = d.$$

The first condition requires the hypothesis set to have stabilized: after observing day Y, no additional hypothesis is eliminated. The second requires the remaining sequence to contain at least one instance of every DCS.

Together, these conditions guarantee that every DCS admits a unique prediction under the surviving hypotheses. For any $d \in \mathcal{D}$, consider a post-convergence day t on which $d_t = d$. Because no hypothesis is eliminated after Y, every $\theta \in \mathcal{R}_Y$ must correctly predict the observed outcome on that day:

$$\hat{o}_t(\theta) = o_t \qquad \forall \theta \in \mathcal{R}_Y.$$

D.3 Example: weekday-based activity

True pattern: Daughter Sim paints a picture between 6 AM and 12 PM on Mondays and Fridays.
Question: Will Daughter Sim paint a picture today between 6 AM and 12 PM?
Partial hint: It is related to the day of the week.

Day t	Date	Weekday	Painted?	Remaining hypotheses
–	—	—	—	508
1	Oct. 15	Wed	no	254
2	Oct. 16	Thu	no	190
3	Oct. 17	Fri	yes	32
4	Oct. 18	Sat	no	16
5	Oct. 19	Sun	no	8
6	Oct. 20	Mon	yes	4
7	Oct. 21	Tue	no	2
8–14	Oct. 22–28	Wed … Tue	Fri, Mon: yes	2

Table 18: Elimination trace for the example chain. Daughter Sim paints exactly on Mondays and Fridays; the remaining set shrinks from 508 to 2 by day 7. The convergence day is therefore day 7.

Elimination trace. Table 18 tracks $|\mathcal{R}_t|$ as days are observed, starting from the 508 hypotheses established above. Each observed day removes hypotheses whose prediction disagrees with whether Daughter Sim painted that morning. For example, day 1 is a Wednesday with no painting, so hypotheses predicting painting on Wednesday are eliminated. Day 3 is a Friday with painting, so hypotheses predicting no painting on Friday are eliminated.

After all weekdays have been observed once, two hypotheses remain:

$$\theta_A : W = \{\text{Mon}, \text{Fri}\}, \quad \text{paint on days in } W,$$

and

$$\theta_B : \quad W = \{\text{Tue}, \text{Wed}, \text{Thu}, \text{Sat}, \text{Sun}\}, \quad \text{paint on days outside } W.$$

These hypotheses use different symbolic descriptions, but they make identical predictions: both imply that Daughter Sim paints exactly on Mondays and Fridays. Therefore, when the direct question is asked after convergence, e.g. on Friday, Oct. 24, the unique answer is *yes*.

E Dialogue Generation and Adaptation

The Sims 4 recordings used in SimLife contain silent "simlish" mouth animations rather than audible speech. We therefore synthesize English dialogue after recording. The dialogue pipeline has two purposes:

- **Original dialogue generation**: makes each chat interaction sound natural and temporally aligned with the video.
- **Prompted dialogue adaptation**: modifies generated dialogue to introduce phenomenon-relevant topics, mention activities that are not directly visible, or indicate the change-point event on the pivot day of a dynamic episode.

E.1 Original Dialogue Generation

Chatting session detection. Original dialogue generation starts from the trimmed gameplay video and game action log. The action log is used to identify chatting sessions, defined as contiguous intervals in which two or more Sims are plausibly engaged in the same log-recorded chat event. Failed, canceled, idle, and menu-level actions are removed, and overlapping chat actions among nearby Sims are grouped into sessions. Sessions are merged only when participants remain temporally and spatially close, and split when a participant leaves or begins a separate chat with someone outside the session.

Visual speech refinement. Because log-derived action boundaries are often coarser than visible speech, we refine each session using mouth-motion cues from the video. Faces are detected in sampled frames, assigned to characters using reference images, and analyzed with lip landmarks. We

compute a mouth aspect ratio,

$$\mathrm{MAR} = \frac{\frac{1}{3}\sum_{(i,j)\in P}\|\mathbf{x}_i - \mathbf{x}_j\|}{\|\mathbf{x}_{\mathrm{right}} - \mathbf{x}_{\mathrm{left}}\|}, \tag{10}$$

where P contains inner-lip landmark pairs and the denominator is the mouth-corner distance. Frames with sufficiently large mouth opening are treated as visual speech, while wide-smile cases are suppressed using mouth-width geometry. The visual speech intervals are then fused with the action-log session boundaries.

Speaker timeline construction. For each fused session, we construct a speaker timeline: an ordered sequence of speaker slots with start time, end time, speaker identity, and word budget. Visual speech intervals receive highest priority, while action-log rows provide softer evidence for who is likely speaking. Remaining gaps are filled by rotating among active participants, and very short slots are constrained to brief interjections.

Dialogue generation. We generate dialogue in two model-assisted steps using Qwen3.5-35B-A3B. First, the model captions the session clip, describing visible participants, facial expressions, mouth motion, interaction context, environment, and mood from Servo Bot's first-person viewpoint. Second, the model is prompted with the caption, speaker timeline, activity context, and character profiles to generate exactly one utterance per slot.

Speech synthesis and audio mixing. Each utterance is synthesized using Qwen3-TTS-12Hz-1.7B-CustomVoice[5] with character-specific voice assignments and mixed into the video. Speech is speed-adjusted when necessary to preserve temporal alignment. Speaker volume is modulated by the distance to Servo Bot:

$$v(d) = \mathrm{clamp}\left(1 - \frac{d}{25}, 0, 1\right), \tag{11}$$

with a soft lower gain floor to keep distant speech weak but audible.

E.2 Prompted Dialogue Adaptation

After producing the original dialogue track, we adapt selected sessions to support phenomena whose target rules cannot be fully inferred from visible frames alone. Dialogue adaptation is controlled through additional prompt instructions in the dialogue generation stage.

Invisible activity completion. Some phenomenon-relevant activities are not directly visible because they occur outside the camera view or during occlusion. In these cases, dialogue can make the missing activity inferable by referring to activities that have already happened, are currently being planned, or are about to happen. Such references must be consistent with the underlying action log and schedule, and are restricted to same-day activities to avoid revealing underlying patterns.

Dynamic Episodes. For *dynamic episodes*, the household follows one pattern before a pivot day and another pattern after it. The pivot day contains a change-point event that explains or motivates this shift. During prompted dialogue adaptation, we inject this event into one of the pivot-day conversations without explicitly stating the full before–after rule. For example, a pivot-day dialogue session involving Mother Sim may mention that she is preparing for an art market, causing her to switch from painting every other day to painting every day.

E.3 Constraints on Adapted Dialogue

Prompted adaptation must remain temporally aligned with the original speaker timeline, visually and causally consistent with the recorded video, action log, and character profiles, and must not directly reveal the benchmark answer. The dialogue may provide clues, reminders, or references to relevant events, but the target phenomenon must still be inferred from the full multimodal context.

[5] https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-CustomVoice

E.4 Quality Control

We apply quality control at both the model-generation stage and the human-inspection stage. During original dialogue generation, the session caption is first generated from the video clip. The dialogue-generation model then receives this caption together with the speaker timeline, activity context, and game log information. Before generating the final utterances, the model is prompted to check whether the caption is consistent with the log-derived session context, including visible participants, activities, and temporal ordering.

After dialogue generation, we further inspect generated sessions using our visualization platform. We conduct manual spot checks to verify that the dialogue is temporally aligned, visually plausible, and consistent with the underlying activity logs. This inspection is used to identify obvious captioning errors, speaker mismatches, timing issues, or dialogue that reveals more information than intended.

F SimLife Web Interface

The following sections describe the features and use of the **SimLife** platform.

We include three web interfaces to facilitate the data creation pipeline for **SimLife-BP**:

- **Simulated Day Timeline:** Assists in parsing raw JSON schedules by combining individual characters' assigned schedules into a unified, side-by-side family timeline. This visual verification step ensures game mods correctly enforce scheduled actions.
- **Phenomenon Generation:** Provides an end-to-end platform to construct behavioral benchmarks. This page runs simulated day sampling, evaluates activity visibility each day, and verifies hypothesis elimination before adding video chains to the database.
- **Task Annotation:** A platform for human validation of data and ground-truth question insertion. Video playback is shown alongside character game logs to identify the right frame to insert the benchmark questions.

F.1 Simulated Day Timeline

We created the "Simulated Day Timeline" page, as shown in Figure 5, to view the simulated day schedules we generated for each character. Each block on the day schedule is an activity that we assigned to the character and we use game mods to ensure that the activity will happen around the time it is scheduled.

Since we store the schedules in JSON files with each character's assigned activities as a list, it is difficult to visualize what the family as a whole will be doing in a single day when we run the game recording pipeline. This web interface allows us to verify if activities happen when we want them to, which is within the scheduled start and end time block, and also have a better idea of how the schedules of each family member will interact.

F.2 Phenomenon Generation

We use the Phenomenon Generation page to create the phenomenon, as described in Appendix B, and then generate video chains that fulfill the requirements of the phenomenon. This page allows us to build a phenomenon piece by piece while validating that it is possible to create a video chain data instance and adding new video chains to the dataset through an automatic simulated day video selection process.

Validating Phenomenon The main utility from this page comes after clicking the "Validate" button for a phenomenon, where the backend will first perform the simulated day sampling outlined in Appendix B to create a video chain with the number of days and start date from the phenomenon. The days will be shown in a "Schedule Calendar". Figure 7 is the schedule calendar for a dynamic understanding phenomenon, where you can see that there is a purple "EVO" day, which represents a day separating the two human profiles and the day that a trigger event should be inserted.

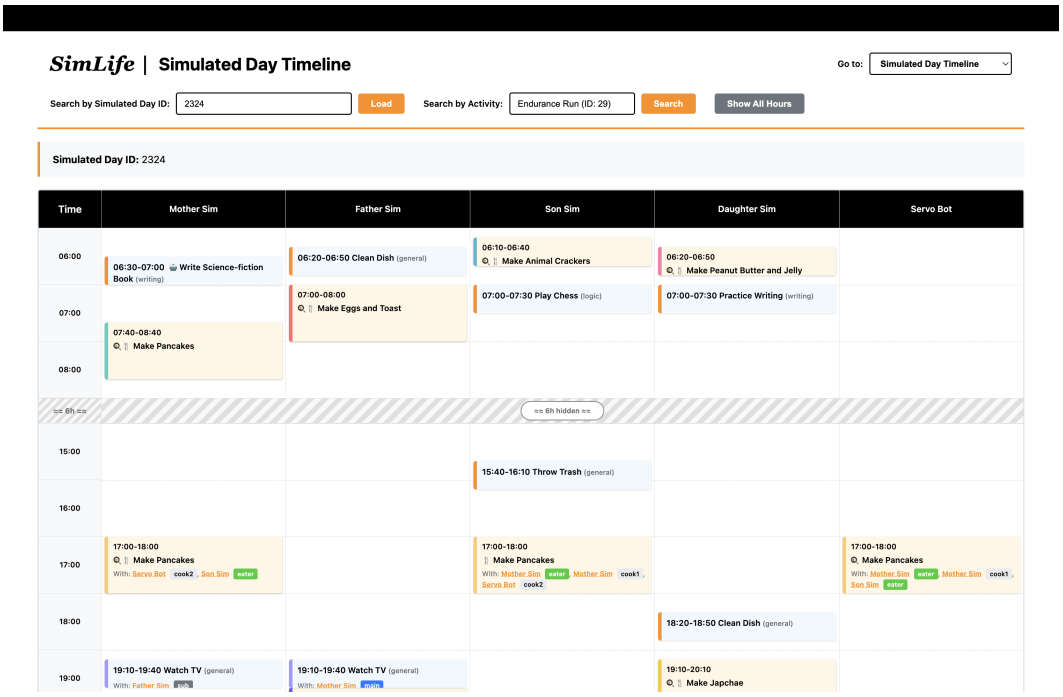

Figure 5: Timeline interface that displays the scheduled actions for a specified simulated day for each character in the game. This page will display the schedules the Sims characters will perform when we run the specified simulated day in the game.

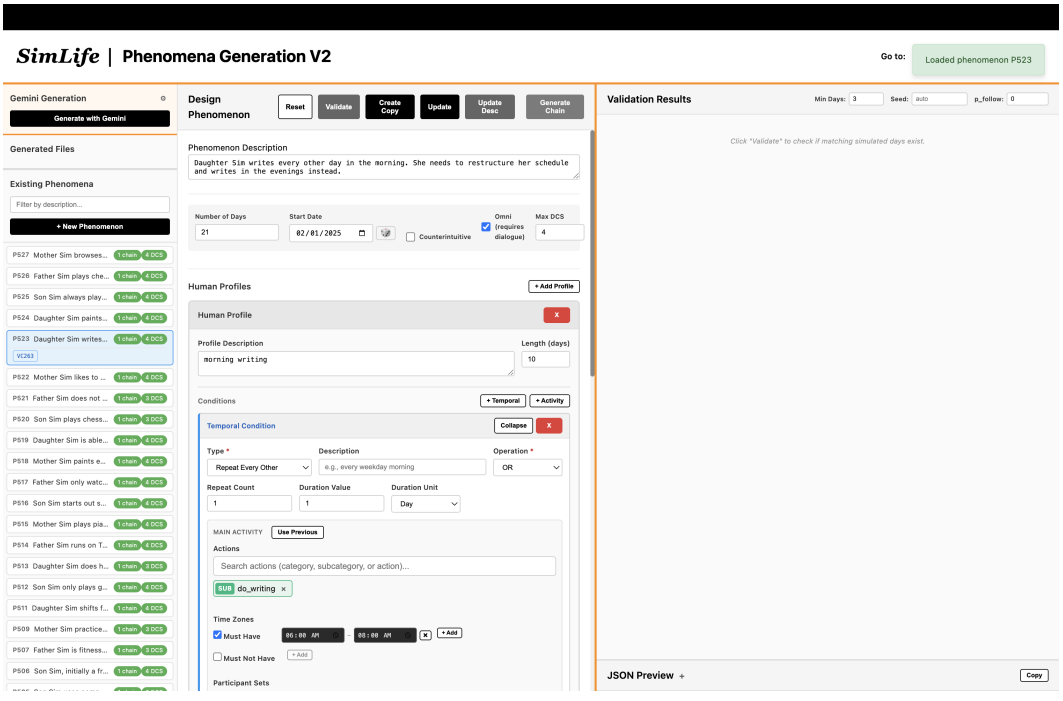

Figure 6: Phenomenon Generation page from the SIMLIFE platform. This is what is shown when selecting a phenomenon from the left sidebar, before validating it.

The activity sets shown below are the simulated day IDs that fulfill the condition requirements and the visibility score of each day. The first schedule calendar shown will always select the days with the highest visibility first.

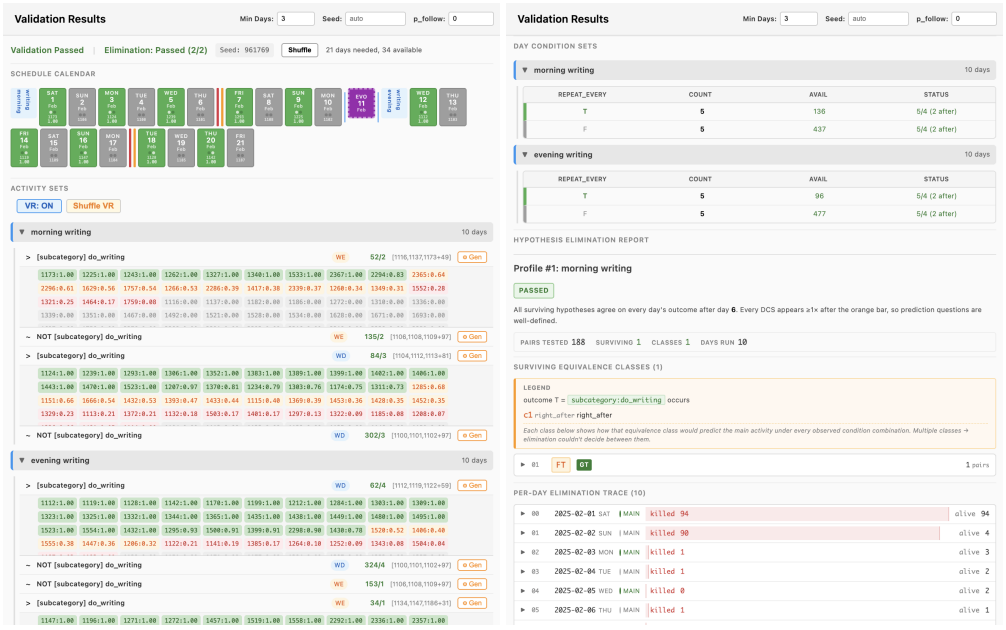

Figure 7: Output displayed on the Phenomenon Generation page after validation of a phenomenon. **Left:** The calendar for the specific video chain and available simulated days for a condition set. **Right:** Scrolling down will show the truth tables for day condition sets and hypothesis elimination trace.

There are also two lines per human profile shown on the schedule calendar: the red line indicates when there are at least 3 occurrences for each day condition set, while the orange line indicates when hypothesis elimination passes. Either of these processes can fail, for example, there may not be enough simulated days to fill out the entire schedule calendar or hypothesis elimination isn't possible within the number of days for this schedule. The "Generate Chain" button will only be available when both pass and there is a red and orange line with at least one occurrence of each day condition set truth table assignment after that point.

Then, clicking on "Generate Chain" will add the exact video chain with the simulated day IDs shown in the schedule calendar to the database.

F.3 Task Annotation

The task annotation page allows us to insert tasks at a specific time in a video chain while watching the video to ensure visibility of activities. This page enables human validation of the video chain data instances that are generated from the phenomenon and easy insertion of tasks since all information is consolidated in one view.

Day Summary The day summary includes all actions from the game logs, split by character. Each one is named and grouped based on the category of action. The green bar next to an action name is the visibility score of each activity. Clicking an activity will jump directly to the start time of that action in the video. This function can be used to identify the point in time in the video that is after a certain activity to find a place to insert a task.

Inserting Tasks The "Task Annotation" section on the right of the video in Figure 8 allows us to write a task question, select the type of question and correct answer, and label it with a simulated day and timestamp of the video which will be the end of the context provided to a model before the task question will be presented.

Scrolling down on this section will show an interface to fill out information about the task question to be inserted.

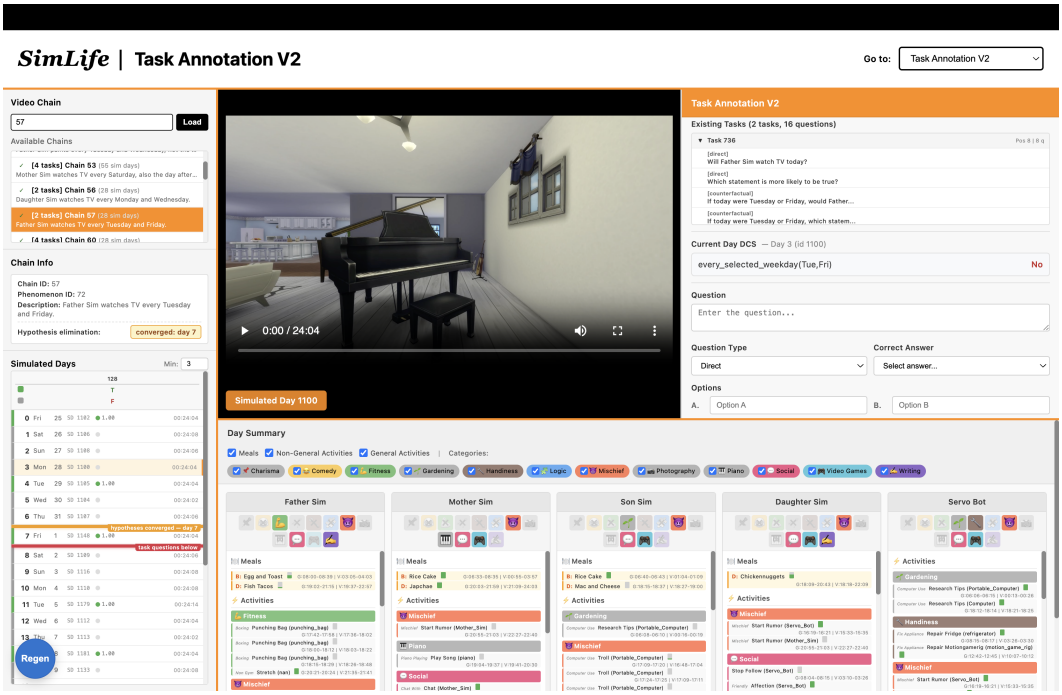

Figure 8: Task Annotation page from the **SIMLIFE** platform. This is the appearance after selecting a video chain and a specific simulated day within the video chain in the left sidebar.

Code	Type	Description	Days
A0	Anchor (easy)	Mother Sim plays piano in the evening every other day.	10
A1	Anchor (hard)	Daughter Sim writes everyday without painting in the same day.	20
N0	Intuitive (temporal)	Mother Sim paints a canvas every Tuesday.	28
N1	Intuitive (activity)	Father Sim always drinks coffee the day after playing chess at night.	21
N2	Counterintuitive (activity)	Father Sim will do fitness and fishing in the same day.	21
N3	Counterintuitive (mixed)	Daughter Sim writes screenplays, is inspired immediately after watching TV.	16
N4	Dynamic	Father Sim runs on Tuesday and Thursday. He wants to participate in a half marathon and starts training every other day.	29
N5	Dynamic	Mother Sim paints every other day, then in order to prep for an art market, begins to paint every single day.	29
N6	Dynamic	Son Sim plays chess late at night every day. As he needs to wake up early in the morning due to a new club on weekdays, he only practices chess late at night on weekends.	25
N7	Dynamic	Daughter Sim is able to play motion games more often at the beginning of the school year, but as she gets more busy, plays motion games only on the weekends.	29

Table 19: Overview of the episodes used in the human study.

G HUMAN STUDY

We conduct the human study through a web-based interface. The study uses a subset of 10 video chains, listed in Table 19. Each participant completes two shared anchor episodes, `anchor0` and `anchor1`, for calibration (Section G.2); the remaining episodes are distributed so that each is evaluated by three participants.

Each episode presents a household of four Sims over multiple simulated days. On each day, participants view a daily activity calendar with visible and hidden activity blocks, answer factual questions from the visible calendar, and rate their confidence about whether a target activity occurred. Participants first complete an interactive tutorial introducing the task, interface, and response format.

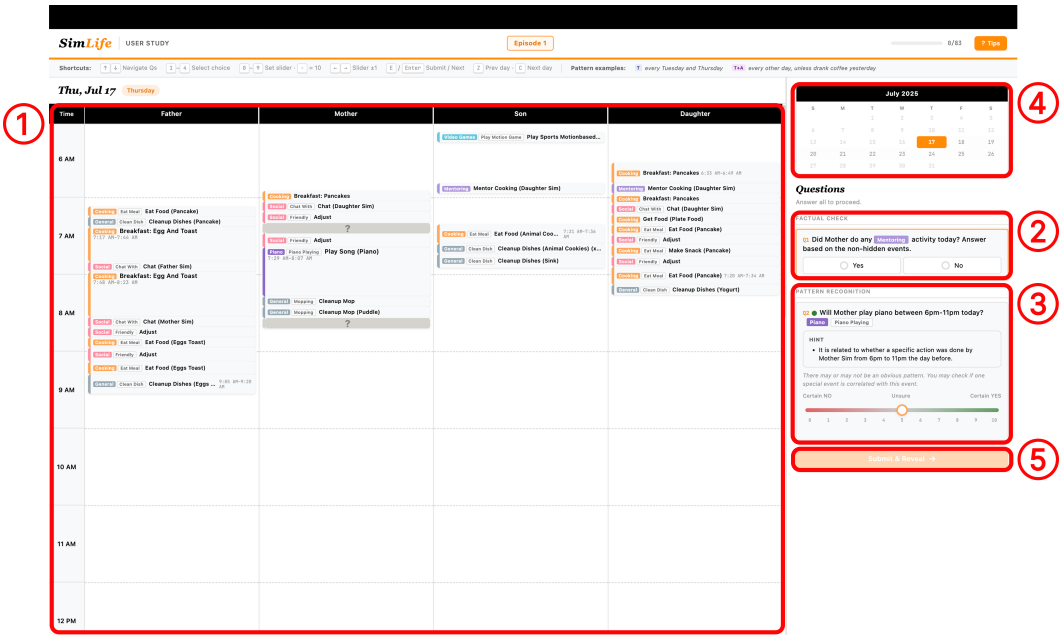

Figure 9: Overview of the human study interface. Some activity blocks are hidden from participants and shown as gray blocks.

G.1 User Interface

Figure 9 shows the main interface. The daily calendar (marked 1 in Figure 9) displays each Sim's activities for the current day, with hidden activities shown as unknown blocks. These hidden blocks may include the target pattern activity or other contextual activities.

For each day, participants answer factual questions (2) and pattern-prediction questions (3). Factual questions check attention to the visible schedule. Pattern-prediction responses are collected on an 11-point scale from 0 ("certainly no") to 10 ("certainly yes"). Each prediction question is accompanied by a partial hint, described in Section 3.2, which identifies a relevant class of evidence without revealing the full rule. For dynamic chains, a pop-up window notifies participants of the pattern change on the pivot day. A compact history view in the form of a calendar (4) summarizes whether the target activity occurred on previous days. After submission (5), hidden blocks are revealed and the target activity is highlighted when present, allowing participants to update their hypotheses over the episode.

G.2 Anchor-Based Participant Calibration

Participants may differ in learning speed and slider usage. To adjust for these participant-level effects, each participant completes two fixed anchor tasks designed to differ in difficulty: an easy anchor A_0 (*"Mother plays piano every other evening"*) and a hard anchor A_1 (*"Daughter writes on days when she does not paint"*).

On each day d, participant p reports an integer slider response $\text{slider}_{p,d} \in \{0, \ldots, 10\}$ indicating whether the target activity occurs on that day. We interpret this response as a probability $\text{slider}_{p,d}/10 \in [0, 1]$, and denote the ground truth by $y_d \in \{0, 1\}$. We fit a participant-specific slider mapping on the anchor responses:

$$\frac{\text{slider}_{p,d}}{10} = a_p\, y_d + b_p + \varepsilon_{p,d},$$

where a_p captures discriminability, i.e., the slider range used to separate $y_d = 1$ from $y_d = 0$ days, b_p captures base-rate bias, i.e., the expected response on $y_d = 0$ days, and $\varepsilon_{p,d}$ captures residual noise.

We also estimate a participant-specific learning-speed offset from anchor convergence days:

$$D_{\text{conv}}(p, t) = \mu_t + \alpha_p + \varepsilon_{p,t},$$

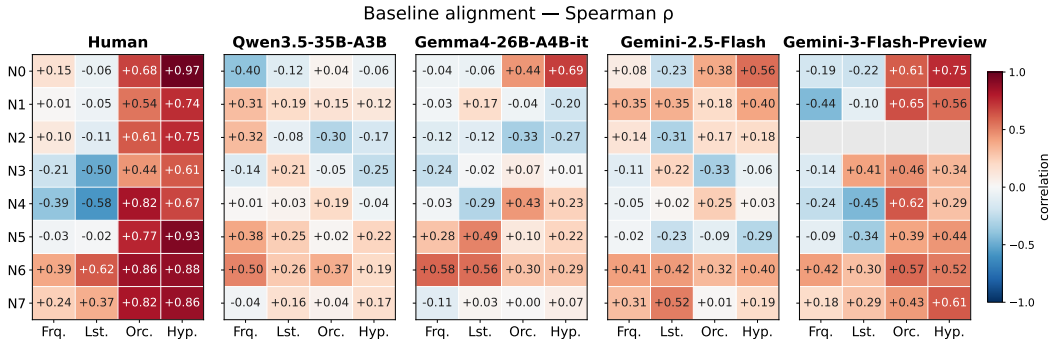

Figure 10: Baseline alignment of day-level prediction trajectories (Spearman ρ). Each panel corresponds to one path that produces a daily $P(\text{yes})$ trajectory for the target event: Human (per-person consolidated mean over the 12 participants, anchor-calibrated), Qwen3.5-35B-A3B, Gemma-4-26B-A4B-it, Gemini-2.5-Flash, and Gemini-3-Flash-Preview. Within a panel, each cell is the Spearman correlation between that path's trajectory and one of four reference baselines, computed over the days of a case. Rows N0–N7 are the eight non-anchor cases. Columns are the baselines: Frq. = frequency-based, predicting from the historical base rate; Lst. = last-day-based, extrapolating from the previous day's outcome; Orc. = ground-truth oracle; Hyp. = hypothesis-elimination, pruning candidate rules against accumulated evidence. A cell is undefined (shown in grey) when the source's prediction is constant across that case's days, so the rank correlation has zero variance. Gemini-3-Flash-Preview emits only binary 0/1 probabilities and answers $P(\text{yes}) = 0$ on every one of the 21 days of N2 ("Father fitness + fishing same day"), never predicting "yes"; consequently all four Gemini-3 cells in row N2 are undefined and shown grey. Spearman ρ is rank-based (amplitude-free) and robust to the models' tendency to saturate at 0 and 1.

where $D_{\text{conv}}(p,t)$ is the convergence day of participant p on anchor task t, μ_t is the mean convergence day for task t (see D.2), and α_p captures whether participant p tends to converge earlier or later than average. We fit both calibration components using hierarchical models, so noisy participant-level estimates are shrunk toward the group mean.

The estimated parameters are then applied to each participant's non-anchor tasks. Each slider response is converted into a calibrated probability:

$$\hat{p}_{p,t,d} = \text{clip}\big((\text{slider}_{p,t,d}/10 - b_p)/a_p,\, 0, 1\big),$$

and the day index is adjusted by the participant's learning-speed offset:

$$\tilde{d} = d - \alpha_p.$$

We then average calibrated curves across participants. The anchor tasks are excluded from the main human–model comparison.

G.3 Results

We present the human study results with representative models in Figures 10 and 11. In general, human participants perform well on both the intuitive cases (N0 and N1) and the dynamic cases (N4-N7).

H Additional Results

We present additional experiment results in this section. Figure 10 shows Spearman correlation between the non-anchor chains used in the human study and the reference baselines for each model evaluated in RQ2 and the human study responses. Figure 11 shows the trajectories for non-anchor cases not mentioned in the body of the paper. Table 20 is the full results for RQ3 on the dynamic understanding video chains.

For all models used in this work, we strictly follow their licenses and terms of use.

The detailed version of the main result in Table 1 is shown in Table 21.

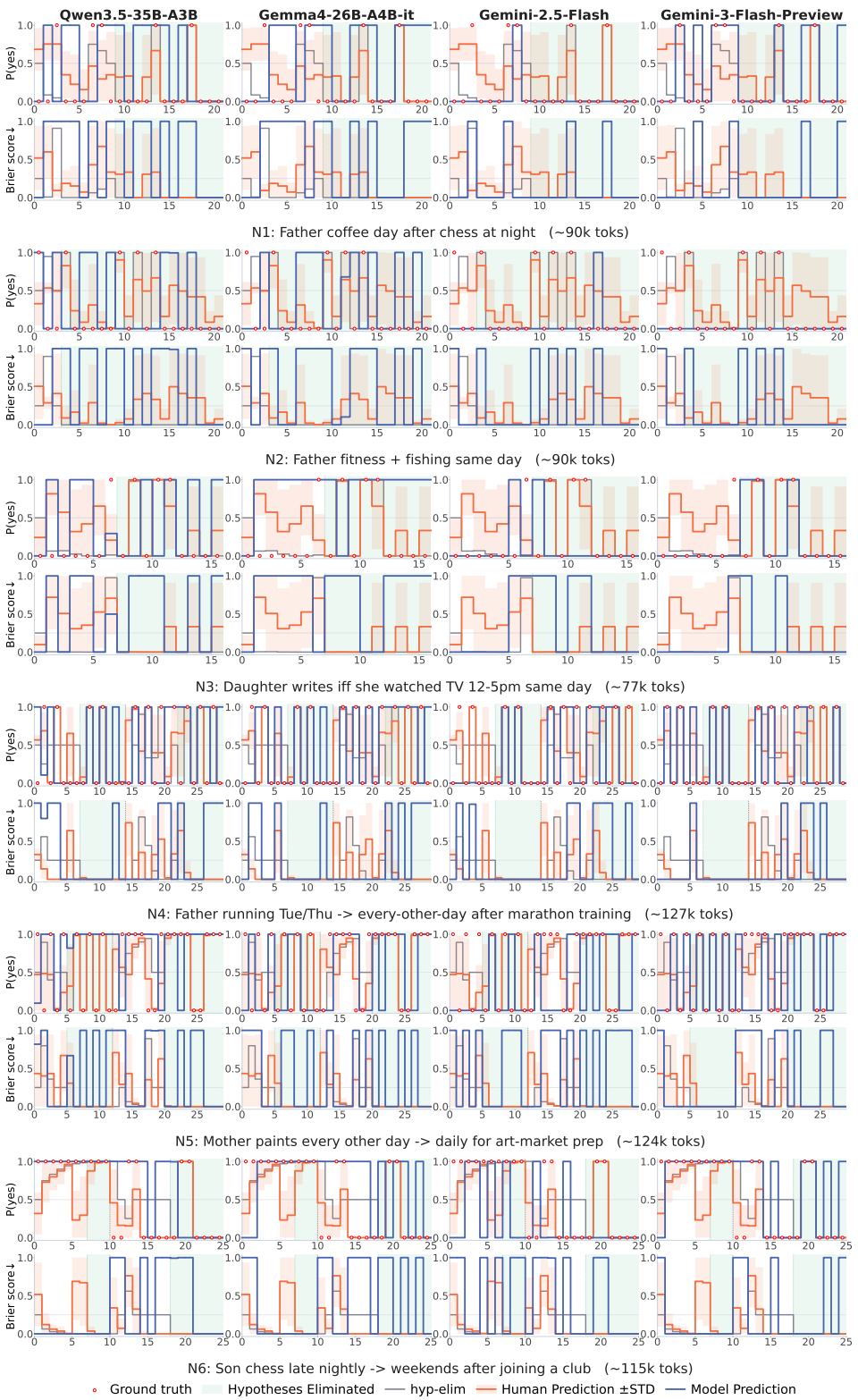

Figure 11: Additional per-day trajectories of model predictions and human judgments for other non-anchor cases (N1-N6). A rapid convergence of Brier score (bottom) toward zero after hypothesis elimination indicates successful pattern understanding and retention.

Model	FPS	Thinking	Full Hints			Partial Hints			No Hints		
			P1	P2	Hist	P1	P2	Hist	P1	P2	Hist
Vision + Language (V+L)											
Qwen3-VL-4B-Instruct	0.5	CoT	$54.5^{\pm 8.1}$	$56.7^{\pm 7.9}$	$55.1^{\pm 6.6}$	$57.2^{\pm 8.1}$	$55.3^{\pm 8.0}$	$50.9^{\pm 6.7}$	$56.6^{\pm 8.1}$	$54.0^{\pm 8.0}$	$55.6^{\pm 6.6}$
Qwen3-VL-8B-Instruct	0.5	CoT	$55.9^{\pm 8.1}$	$47.3^{\pm 8.0}$	$55.6^{\pm 6.6}$	$59.3^{\pm 8.0}$	$60.0^{\pm 7.8}$	$54.2^{\pm 6.6}$	$56.6^{\pm 8.1}$	$54.7^{\pm 8.0}$	$50.5^{\pm 6.7}$
Qwen3-VL-32B-Instruct	0.5	CoT	$57.9^{\pm 8.0}$	$54.7^{\pm 8.0}$	$55.6^{\pm 6.6}$	$59.3^{\pm 8.0}$	$60.0^{\pm 7.8}$	$56.5^{\pm 6.6}$	$55.2^{\pm 8.1}$	$56.7^{\pm 7.9}$	$56.9^{\pm 6.6}$
Qwen3.5-9B	0.5	R	$49.0^{\pm 8.1}$	$50.7^{\pm 8.0}$	$57.4^{\pm 6.6}$	$55.9^{\pm 8.1}$	$57.3^{\pm 7.9}$	$56.5^{\pm 6.6}$	$49.7^{\pm 8.1}$	$55.3^{\pm 8.0}$	$56.0^{\pm 6.6}$
Qwen3.5-35B-A3B	0.5	R	$53.8^{\pm 8.1}$	$48.0^{\pm 8.0}$	$57.9^{\pm 6.6}$	$57.9^{\pm 8.0}$	$56.7^{\pm 7.9}$	$57.9^{\pm 6.6}$	$56.6^{\pm 8.1}$	$55.3^{\pm 8.0}$	$54.2^{\pm 6.6}$
Gemma-4-26B-A4B-it	0.5	R	$49.0^{\pm 8.1}$	$54.7^{\pm 8.0}$	$47.2^{\pm 6.7}$	$54.5^{\pm 8.1}$	$49.3^{\pm 8.0}$	$45.8^{\pm 6.6}$	$56.6^{\pm 8.1}$	$52.0^{\pm 8.0}$	$49.5^{\pm 6.7}$
Gemini-3-flash-preview	0.5	R	$51.7^{\pm 8.1}$	$48.7^{\pm 8.0}$	$57.4^{\pm 6.6}$	$57.9^{\pm 8.0}$	$56.7^{\pm 7.9}$	$55.6^{\pm 6.6}$	$56.6^{\pm 8.1}$	$55.3^{\pm 8.0}$	$55.6^{\pm 6.6}$
VideoTree (Mixtral-8x7B)	1	CoT	$55.9^{\pm 8.1}$	$52.0^{\pm 8.0}$	$57.9^{\pm 6.6}$	$48.3^{\pm 8.1}$	$51.3^{\pm 8.0}$	$59.3^{\pm 6.6}$	$54.5^{\pm 8.1}$	$54.0^{\pm 8.0}$	$55.1^{\pm 6.6}$
Vision + Audio (V+A)											
Qwen3-Omni-30B-A3B + Qwen3.5-9B	0.5	R	$45.5^{\pm 8.1}$	$46.7^{\pm 8.0}$	$51.9^{\pm 6.7}$	$51.0^{\pm 8.1}$	$51.3^{\pm 8.0}$	$53.7^{\pm 6.6}$	$50.3^{\pm 8.1}$	$50.0^{\pm 8.0}$	$49.5^{\pm 6.7}$
M3-Agent	1	R	$49.0^{\pm 8.1}$	$55.3^{\pm 8.0}$	$57.9^{\pm 6.6}$	$60.7^{\pm 8.0}$	$56.7^{\pm 7.9}$	$57.9^{\pm 6.6}$	$53.1^{\pm 8.1}$	$57.3^{\pm 7.9}$	$58.8^{\pm 6.6}$
Language Only (L)											
Qwen3.5-9B	–	R	$48.3^{\pm 8.1}$	$45.3^{\pm 8.0}$	$50.5^{\pm 6.7}$	$54.5^{\pm 8.1}$	$42.7^{\pm 7.9}$	$51.9^{\pm 6.7}$	$63.4^{\pm 7.8}$	$51.3^{\pm 8.0}$	$53.7^{\pm 6.6}$
Qwen3.5-35B-A3B	–	R	$71.0^{\pm 7.4}$	$51.3^{\pm 8.0}$	$69.0^{\pm 6.2}$	$63.4^{\pm 7.8}$	$48.7^{\pm 8.0}$	$65.3^{\pm 6.3}$	$62.8^{\pm 7.9}$	$54.0^{\pm 8.0}$	$63.0^{\pm 6.4}$
Gemma-4-26B-A4B-it	–	R	$64.1^{\pm 7.8}$	$45.3^{\pm 8.0}$	$53.7^{\pm 6.6}$	$66.2^{\pm 7.7}$	$51.3^{\pm 8.0}$	$51.9^{\pm 6.7}$	$68.3^{\pm 7.6}$	$47.3^{\pm 8.0}$	$50.0^{\pm 6.7}$
Gemini-2.5-flash	–	R	$69.7^{\pm 7.5}$	$53.3^{\pm 8.0}$	$59.7^{\pm 6.5}$	$67.6^{\pm 7.6}$	$47.3^{\pm 8.0}$	$63.9^{\pm 6.4}$	$57.9^{\pm 8.0}$	$52.7^{\pm 8.0}$	$63.4^{\pm 6.4}$
Gemini-3-flash-preview	–	R	$82.1^{\pm 6.2}$	$63.3^{\pm 7.7}$	$71.8^{\pm 6.0}$	$86.9^{\pm 5.5}$	$61.3^{\pm 7.8}$	$74.5^{\pm 5.8}$	$75.9^{\pm 7.0}$	$64.0^{\pm 7.7}$	$74.5^{\pm 5.8}$
Gemini-3.1-flash-lite	–	R	$71.7^{\pm 7.3}$	$52.7^{\pm 8.0}$	$65.7^{\pm 6.3}$	$75.2^{\pm 7.0}$	$48.7^{\pm 8.0}$	$59.3^{\pm 6.6}$	$60.7^{\pm 8.0}$	$52.0^{\pm 8.0}$	$64.8^{\pm 6.4}$
Language only, noiseless (L-N)											
Qwen3.5-9B	–	R	$80.7^{\pm 6.4}$	$56.0^{\pm 7.9}$	$71.8^{\pm 6.0}$	$73.1^{\pm 7.2}$	$58.0^{\pm 7.9}$	$69.0^{\pm 6.2}$	$68.3^{\pm 7.6}$	$50.0^{\pm 8.0}$	$66.7^{\pm 6.3}$
Qwen3.5-35B-A3B	–	R	$85.5^{\pm 5.7}$	$72.7^{\pm 7.1}$	$79.6^{\pm 5.4}$	$85.5^{\pm 5.7}$	$65.3^{\pm 7.6}$	$77.3^{\pm 5.6}$	$74.5^{\pm 7.1}$	$55.3^{\pm 8.0}$	$72.7^{\pm 5.9}$
Gemma-4-26B-A4B-it	–	R	$81.4^{\pm 6.3}$	$73.3^{\pm 7.1}$	$72.7^{\pm 5.9}$	$73.1^{\pm 7.2}$	$72.0^{\pm 7.2}$	$72.7^{\pm 5.9}$	$74.5^{\pm 7.1}$	$64.7^{\pm 7.6}$	$72.2^{\pm 6.0}$
Gemma-4-31B-it	–	R	$84.1^{\pm 5.9}$	$72.7^{\pm 7.1}$	$78.2^{\pm 5.5}$	$82.1^{\pm 6.2}$	$64.7^{\pm 7.6}$	$77.8^{\pm 5.5}$	$84.1^{\pm 5.9}$	$67.3^{\pm 7.5}$	$76.4^{\pm 5.7}$
Gemini-2.5-flash	–	R	$86.9^{\pm 5.5}$	$68.0^{\pm 7.5}$	$73.6^{\pm 5.9}$	$85.5^{\pm 5.7}$	$57.3^{\pm 7.9}$	$67.6^{\pm 6.2}$	$82.1^{\pm 6.2}$	$62.0^{\pm 7.8}$	$70.8^{\pm 6.1}$
Gemini-3-flash-preview	–	R	$77.2^{\pm 6.8}$	$62.0^{\pm 7.8}$	$77.8^{\pm 5.5}$	$79.3^{\pm 6.6}$	$65.3^{\pm 7.6}$	$76.4^{\pm 5.7}$	$75.9^{\pm 7.0}$	$62.0^{\pm 7.8}$	$76.9^{\pm 5.6}$

Table 20: Pattern-adaptation breakdown on the **SIMLIFE** dynamic chains (20 chains; question-level accuracy (%) with 95% binomial CI half-width ±). **P1/P2**: normal tasks whose queried day falls before/after the pattern switch (answering about the active regime); **Hist**: the history tasks, which query Pattern-1 behavior after the model has already observed Pattern 2. Unanswered or overflow responses count as incorrect. Full-context language models (L, L-N) generally show P1 > Hist > P2, whereas vision, retrieval, and memory baselines remain near chance with no such structure.

Model	FPS	Thinking	Full Hints						Partial Hints						No Hints					
			AC	Acc	Dir.	CF	N-CF	I-CF	AC	Acc	Dir.	CF	N-CF	I-CF	AC	Acc	Dir.	CF	N-CF	I-CF
Vision + Language (V+L)																				
Qwen3-VL-4B-Instruct	0.5	CoT	$10.5^{\pm 3.1}$	$49.8^{\pm 2.6}$	46.6	55.1	48.6	48.1	$11.5^{\pm 3.2}$	$50.0^{\pm 2.6}$	48.9	52.8	50.4	47.8	$9.4^{\pm 2.9}$	$50.3^{\pm 2.6}$	50.5	53.3	49.6	47.8
Qwen3-VL-8B-Instruct	0.5	CoT	$12.6^{\pm 3.3}$	$50.1^{\pm 2.6}$	46.3	54.1	48.3	51.1	$9.7^{\pm 3.0}$	$48.6^{\pm 2.6}$	44.3	51.4	49.1	48.9	$7.6^{\pm 2.7}$	$48.0^{\pm 2.6}$	48.2	49.9	48.3	45.7
Qwen3-VL-32B-Instruct	0.5	CoT	$11.0^{\pm 3.1}$	$51.9^{\pm 2.6}$	54.7	53.5	50.7	49.2	$13.6^{\pm 3.4}$	$53.2^{\pm 2.6}$	53.1	53.0	53.0	53.8	$12.1^{\pm 3.3}$	$53.0^{\pm 2.6}$	51.8	55.6	52.8	51.4
Qwen3.5-9B	0.5	R	$12.3^{\pm 3.3}$	$48.9^{\pm 2.6}$	47.2	52.0	50.1	45.9	$14.2^{\pm 3.5}$	$51.4^{\pm 2.6}$	51.1	53.8	52.0	48.4	$13.4^{\pm 3.4}$	$51.0^{\pm 2.6}$	52.4	54.1	50.7	47.0
Qwen3.5-35B-A3B	0.5	R	$13.1^{\pm 3.4}$	$52.2^{\pm 2.6}$	50.2	56.7	53.3	48.1	$14.2^{\pm 3.5}$	$53.3^{\pm 2.6}$	53.7	55.6	50.7	53.2	$12.3^{\pm 3.3}$	$52.5^{\pm 2.6}$	53.4	55.4	47.8	53.5
Gemma-4-26B-A4B-it	0.5	R	$7.6^{\pm 2.7}$	$48.8^{\pm 2.6}$	50.8	47.8	45.7	51.4	$7.6^{\pm 2.7}$	$48.1^{\pm 2.6}$	50.5	50.9	45.7	45.7	$10.8^{\pm 3.1}$	$49.3^{\pm 2.6}$	50.2	53.0	45.7	48.4
Gemini-3-flash-preview	0.5	R	$14.7^{\pm 3.6}$	$53.0^{\pm 2.6}$	53.7	59.3	52.2	46.5	$13.9^{\pm 3.5}$	$54.1^{\pm 2.6}$	56.7	57.2	54.3	48.6	$14.2^{\pm 3.5}$	$54.8^{\pm 2.6}$	53.7	56.2	55.9	53.0
VideoTree (Mixtral-8x7B)	1	CoT	$11.0^{\pm 3.1}$	$54.0^{\pm 2.6}$	45.9	56.2	54.6	57.8	$9.7^{\pm 3.0}$	$51.7^{\pm 2.6}$	46.3	53.0	56.4	50.0	$10.8^{\pm 3.1}$	$52.6^{\pm 2.6}$	49.8	52.0	54.6	53.5
Vision + Audio (V+A)																				
Qwen3-Omni-30B-A3B + Qwen3.5-9B	0.5	R	$10.8^{\pm 3.1}$	$48.1^{\pm 2.6}$	45.3	50.9	49.6	45.9	$11.0^{\pm 3.1}$	$48.6^{\pm 2.6}$	48.9	49.9	51.7	44.1	$10.8^{\pm 3.1}$	$48.6^{\pm 2.6}$	47.6	50.1	47.8	48.9
M3-Agent	1	R	$10.0^{\pm 3.0}$	$53.4^{\pm 2.6}$	46.3	55.4	50.7	60.0	$10.2^{\pm 3.0}$	$55.0^{\pm 2.6}$	49.2	57.0	54.3	58.4	$12.1^{\pm 3.3}$	$54.6^{\pm 2.6}$	48.2	55.9	55.9	57.0
Language Only (L)																				
Qwen3.5-9B	–	R	$9.2^{\pm 2.9}$	$50.9^{\pm 2.6}$	46.9	62.5	45.4	48.1	$13.6^{\pm 3.4}$	$54.6^{\pm 2.6}$	55.0	61.4	53.8	48.1	$14.2^{\pm 3.5}$	$54.6^{\pm 2.6}$	50.2	61.9	53.8	51.6
Qwen3.5-35B-A3B	–	R	$18.1^{\pm 3.9}$	$61.1^{\pm 2.5}$	54.1	75.1	53.5	60.3	$18.4^{\pm 3.9}$	$59.3^{\pm 2.5}$	59.3	71.4	51.7	54.6	$16.0^{\pm 3.7}$	$58.7^{\pm 2.5}$	49.5	74.0	53.5	55.9
Gemma-4-26B-A4B-it	–	R	$11.3^{\pm 3.2}$	$53.9^{\pm 2.6}$	52.8	61.7	47.8	53.0	$11.0^{\pm 3.1}$	$54.4^{\pm 2.6}$	58.0	55.6	52.5	52.2	$11.3^{\pm 3.2}$	$54.8^{\pm 2.6}$	56.4	62.2	52.2	48.4
Gemini-2.5-flash	–	R	$17.8^{\pm 3.8}$	$58.6^{\pm 2.5}$	57.3	69.8	49.9	57.0	$21.0^{\pm 4.1}$	$60.2^{\pm 2.5}$	59.3	70.1	54.6	56.5	$16.3^{\pm 3.7}$	$56.4^{\pm 2.6}$	51.1	68.5	50.1	54.9
Gemini-3-flash-preview	–	R	$27.8^{\pm 4.5}$	$69.2^{\pm 2.4}$	72.3	81.1	62.5	**61.4**	**33.3**$^{\pm 4.7}$	$69.4^{\pm 2.4}$	71.0	81.1	64.8	**60.5**	$27.8^{\pm 4.5}$	$66.5^{\pm 2.4}$	60.9	81.1	59.1	**63.8**
Gemini-3.1-flash-lite	–	R	$20.7^{\pm 4.1}$	$59.9^{\pm 2.5}$	59.9	68.0	57.5	54.1	$21.0^{\pm 4.1}$	$61.2^{\pm 2.5}$	57.0	73.5	55.4	57.8	$18.6^{\pm 3.9}$	$59.1^{\pm 2.5}$	52.4	71.9	53.8	56.8
Language only, noiseless (L-N)																				
Qwen3.5-9B	–	R	$23.6^{\pm 4.3}$	$65.5^{\pm 2.5}$	70.4	79.3	62.2	50.5	$18.4^{\pm 3.9}$	$61.2^{\pm 2.5}$	64.5	76.9	57.5	46.2	$16.0^{\pm 3.7}$	$56.5^{\pm 2.6}$	59.0	74.0	51.2	41.9
Qwen3.5-35B-A3B	–	R	$34.9^{\pm 4.8}$	$73.0^{\pm 2.3}$	76.9	84.0	70.9	60.5	$32.0^{\pm 4.7}$	**70.3**$^{\pm 2.4}$	71.7	84.0	66.4	58.9	$21.5^{\pm 4.1}$	$62.7^{\pm 2.5}$	61.2	81.9	56.2	50.8
Gemma-4-26B-A4B-it	–	R	$30.2^{\pm 4.6}$	$72.1^{\pm 2.3}$	82.4	**86.9**	64.8	55.7	$26.0^{\pm 4.4}$	$68.4^{\pm 2.4}$	**75.6**	84.3	62.5	52.2	$22.3^{\pm 4.2}$	$64.1^{\pm 2.5}$	69.4	82.4	54.9	50.5
Gemma-4-31B-it	–	R	**36.2**$^{\pm 4.8}$	**74.6**$^{\pm 2.2}$	**83.4**	86.6	**74.8**	54.9	$30.2^{\pm 4.6}$	$69.6^{\pm 2.4}$	75.2	**86.4**	66.4	51.1	**29.4**$^{\pm 4.6}$	**67.6**$^{\pm 2.4}$	**72.6**	**84.8**	61.7	51.9
Gemini-2.5-flash	–	R	$31.5^{\pm 4.7}$	$71.8^{\pm 2.3}$	80.8	82.9	72.4	52.2	$22.3^{\pm 4.2}$	$65.3^{\pm 2.5}$	70.0	80.8	64.0	46.8	$22.0^{\pm 4.2}$	$63.7^{\pm 2.5}$	64.8	80.3	57.0	52.4
Gemini-3-flash-preview	–	R	$24.1^{\pm 4.3}$	$68.2^{\pm 2.4}$	76.9	81.1	73.5	42.2	$24.9^{\pm 4.3}$	$67.3^{\pm 2.4}$	74.6	80.1	**68.5**	46.8	$23.1^{\pm 4.2}$	$63.9^{\pm 2.5}$	65.5	80.3	**63.8**	45.7

Table 21: Full breakdown of the main results (Table 1) on **SIMLIFE-BP** (381 tasks / 1,439 questions per hint level). AC is the All-Correct rate and Acc the question-level accuracy, both with 95% binomial CI half-widths. The remaining columns give the accuracy on each question type, i.e., the values drawn as bars in Table 1: Dir. = Direct (n=307), CF = Counterfactual (n=381), N-CF = Noise-Counterfactual (n=381), I-CF = Inverse-Counterfactual (n=370), where n is the number of questions of that type per hint level. Shading highlights the within-row best and second-best hint variant for AC and Acc; **bold** marks the overall best in each column.

I Case Study Analysis of Model Reasoning Traces

We examine model reasoning traces to understand how models arrive at their predictions. We have full thinking traces for Gemma and Qwen model families, whereas the Gemini models provide summarized reasoning as part of the final output.

I.1 Hint Levels and Input Modalities

We start by examining the effects of the three hint levels provided in our evaluation on the reasoning of the models.

No Hints. The most common failure, especially for Qwen and Gemini-2.5, is treating the truncated log for the target day as a complete record. Models then simply search that log for the activity. Any use of the history is only to gauge how often it occurs rather than to compare weekdays or find a pattern. As a result, they rarely answer Yes, citing insufficient evidence.

Partial Hints. These direct attention to the specific weekdays where the target activity happens. Models are more likely to build a table of activities by weekday and check whether the target activity matches the day in question. Partial hints also largely eliminate the truncated-log confusion, so part of their performance gain comes from clarifying which day is being predicted.

Full Hints. Models treat the hint as a rule to verify. The cleanest reasoning identifies the current weekday, compares it with the activities named in the hint, and verifies the pattern in the history. A new failure appears with two-condition hints: models check only one condition or read an OR as an AND. Gemini-3-flash-preview uses hints most cleanly. It never confuses the hinted day with the target day, keeps false Yes predictions near zero, and never refers to the hint explicitly; the hint's influence shows only in the variables its reasoning attends to.

Interaction with input modality. Hints will provide the correct variables which determine *which* evidence a model looks for, but the input determines whether that evidence is available:

- **L-N:** The history is complete and structured, so Full Hints reliably produce correct rule application.
- **L:** The evidence exists but is buried in noise. Some models find it (Gemini-3 most consistently); others fall back on a surface reading of the hint.
- **V+L:** The text diaries built from video usually lack the deciding evidence.

Case Study: Task 955, "Will Son Sim play sports motion games between 12pm and 10pm?" With No Hints, every model answers No in every condition. The Partial Hint ("day of the week + an action from 6am to 12pm") is too vague for the L-N setting and the models latch onto the wrong variable. Gemma, for example, links the activity to morning pushups or sports gaming, when the relevant variable is the *absence* of morning family motion gaming. The Full Hint ("Thu/Fri/Sat + motion games 6am–12pm") names the right variable, and every L-N model, along with Gemini-2.5 and Gemini-3 under L, finds the rule. Under V+L, however, no model succeeds with the Full Hint, because the diaries almost never record the three afternoon sessions, meaning there is not enough evidence to apply the rule. Gemini-3's diary contains the only plausible trace (Son exercising in the gaming pod on Day 4), and the model itself notes that this is the sole recorded instance in 17 days.

I.2 Trajectory Probing

Using the RQ2 results, we examine how the reasoning of the four representative models evolves as evidence accumulates.

General patterns. Models typically answer No on Day 1, reflecting the lack of evidence for a recurring pattern. On Day 2, Gemma, Qwen, and Gemini-3 often switch to Yes based on the previous day alone. These predictions are usually incorrect and suggest an overly aggressive attempt to infer a rule from very limited evidence. During the first week, models entertain multiple hypotheses, often

inferred from the specific wording of the hint (e.g., "the day before" or "a specific date" prompts every-other-day or date-of-month rules). Once a weekday repeats, reasoning shifts to comparing the current weekday with past occurrences, which lets models identify static patterns.

Failure modes. We observe three main types of error:

- **Evidence corruption:** inventing observations, missing ones that are present, or attributing another family member's activity to the target Sim.
- **Temporal grounding:** Qwen frequently treats "today" as the previous day, so its predictions track the wrong day. Consistent with this, its second-highest Spearman correlation is with the last-day baseline.
- **Rule updating:** On dynamic patterns, most models keep applying the original rule after behavior changes. Only Gemini-3-flash-preview ever cites the dialogue clue as evidence of a change.

Models can thus accumulate evidence for a recurring pattern but struggle to revise an established rule when new contextual evidence indicates a change. We examine the examples in Figure 4 in detail below.

N0: Mother Sim paints every Tuesday. Mother paints on Days 4, 11, 18, and 25. As expected, no model predicts Day 4, and all four identify the Tuesday pattern around the second occurrence. The main failure is evidence attribution:

- **Gemini-2.5-flash** finds the pattern after Day 11 but but its reasoning contains several hallucinated or misattributed observations. For example, it incorrectly attributes Father's Day 2 painting to Mother and invents a Day 6 session. It then reverts to searching the current day's log instead of applying the pattern, missing Days 18 and 25, whose logs contain only headers.
- **Gemini-3-flash-preview** finds the pattern by Days 11–12 and applies it through the end, but invokes unsupported priors (e.g., that Tuesday and Friday are commonly paired for hobbies) and attributes the children's painting on Days 15 and 22 to Mother, concluding she paints on both Saturdays and Tuesdays.
- **Gemma-4-26B-A4B-it** initially relies on priors (e.g., predicting painting on Day 1 because it is a weekend), but identifies and maintains the Tuesday pattern from Day 10.
- **Qwen3.5-35B-A3B** attributes Father's painting to Mother and predicts No on Day 11, but corrects itself after the second occurrence.

N7: Daughter Sim's motion-game schedule changes. For the first 14 days, Daughter plays motion games Monday through Thursday (Days 1–3, 7–10, and 14). A dialogue clue on Day 15 signals a change, after which she plays only on weekends (Days 19, 20, 26, and 27).

- **Gemini-2.5-flash** initially answers No for lack of evidence and later finds a weekday pattern, but focuses heavily on the current day's log, treating an empty log as evidence that the activity did not occur.
- **Gemini-3-flash-preview** is the only model to use the dialogue clue to recognize the change, correctly reasoning that the schedule has changed. After Day 25, however, it begins proposing unsupported alternatives (e.g., alternating schedules) and misattributing evidence from other family members.
- **Gemma-4-26B-A4B-it** relies on the hint rather than the observed evidence and frequently invents implications from it. It also defaults to Yes and describes Daughter as playing "almost every day,", while never detecting the shift.
- **Qwen3.5-35B-A3B** treats "today" as the previous day on 26 of 29 days, basing predictions on the wrong day.

No model consistently identifies and applies both phases. Models either extrapolate the initial pattern, follow the hint over the evidence, or misread the temporal reference.

J VISION-BASED BASELINE IMPLEMENTATION DETAILS

For models with a vision input channel (Vision + Language and Vision + Audio in Table 1), we use a two-phase pipeline for long-video multiple-choice question answering. Each in-game day (spanning 6:00 AM–10:00 PM game time) is divided into four non-overlapping temporal windows of equal game-time duration. In Phase 1, a vision-language model (VLM) generates a natural-language diary summary for each window from sampled frames. In Phase 2, a text-only model answers the question by reasoning over the concatenated diary summaries. No video frames are shown to the model during Phase 2.

We use vllm 0.19.0 with 1~4 Nvidia A40 GPUs for inference task. It takes between several hours to 7 days to fully run the evaluation for one model.

J.1 TEMPORAL WINDOW DEFINITION

Each simulated day is segmented into four four-hour game-time windows as shown in Table 22.

Key	Phase Label	Game-Time Range
w1	morning	06:00–10:00
w2	midday	10:00–14:00
w3	afternoon	14:00–18:00
w4	evening	18:00–22:00

Table 22: Game-time phase definitions.

Window boundaries in real-video time are recovered by linear interpolation over a set of (video_time_sec, game_time) anchor pairs extracted from the simulation's event log for each recording. For tasks whose question stop-point falls within a window (i.e., tasks that are cut at a partial day), the cut window is re-summarized independently from frames up to the stop time; windows beyond the stop point are omitted entirely.

J.2 PHASE 1: PER-WINDOW DIARY GENERATION

Frame extraction. Frames are sampled at 0.5 fps from the video-time interval corresponding to each window and resized to 224×224 pixels (JPEG quality 85). The resulting JPEG-encoded frames are passed as base64-encoded image_url entries in the prompt. At 0.5 fps, a four-hour window spanning approximately six minutes of real video time yields around 180 frames, though in practice the pipeline caps this at the frames available in the annotated window interval.

Input composition. Each Phase-1 call receives:

- A 2×2 cast image showing the four household members' faces (Father Sim, Mother Sim, Son Sim, Daughter Sim), used to ground character identity.
- The ordered sequence of sampled frames.
- Optionally, a dialogue transcript (in-game HH:MM timestamps) for tasks that require dialogue.
- A text trailer specifying the window label, day number, day-of-week, and date.

System prompt design. The model is framed as the observer whose view constitutes the recorded footage, and is asked to narrate its own movements as "I." Family members are referred to exclusively by role name; no personal names are permitted. The prompt mandates two output sections in strict order: [Detailed actions], a moment-to-moment action list (one fragment per line, $\leq$12 words each, action-verb required), and [Window summary], a prose paragraph not exceeding approximately 250 tokens ($\sim$180 words). Fragments in [Detailed actions] must describe what actors do, not what objects or scenery are present. Good examples include "Mother Sim: stirs pot on stove" and "I: open the fridge"; disallowed examples include "I: see cabinets, bar stools, sink area" and "Father Sim: not visible in this segment." Only the [Window summary] section is carried forward to Phase 2.

Sampling parameters. Four windows are processed concurrently per video day. The model parameters are described in Table 23.

Parameter	Value
Temperature	0.7
Top-p	0.9
Max output tokens	800
Frequency penalty	0.4
Repetition penalty	1.1

Table 23: Phase-1 decoding hyperparameters. Repetition penalty is applied via vLLM's `extra_body` parameter.

Retry strategy. If the model's output fails, and it lacks the [Window summary] section header — typically because the [Detailed actions] list exceeded the token budget — the call is retried up to two additional times (three total tries). On retry, a brief directive is appended to the end of the user message instructing the model to limit [Detailed actions] to 8–12 lines of at most 8 words each, and to emit the [Window summary] header immediately afterward. Retry delays follow a [0, 10, 30] second schedule.

J.3 Phase 2: Text-Only QA over Diary Summaries

Input construction. The Phase-2 prompt presents the concatenated [Window summary] blocks for all days in the video chain, each prefixed with a structured day header (e.g., `Day 3 (Wednesday, 2024-03-06)`), followed by per-window labels (`[morning]`, `[midday]`, etc.). A calendar preamble listing all days with their dates precedes the diary text. The multiple-choice question and options follow the diaries. No images are included.

System prompt. The model is instructed to base its answer exclusively on the provided summaries, to pay attention to day boundaries and recurring patterns, and to produce a response in the format:

```
Thought: [reasoning, max four sentences]
Answer: [letter]
```

Sampling parameters. The model parameters are described in Table 24. The max output tokens is set to be sufficient to cover four sentences of reasoning plus the answer letter.

Parameter	Value
Temperature	0.7
Top-p	0.9
Max output tokens (non-thinking)	512
Max output tokens (thinking enabled)	8704
Frequency penalty	0.4
Repetition penalty	1.1

Table 24: Phase-2 decoding hyperparameters.

Thinking budget (Qwen3/Gemma-4 variants). For models supporting a hybrid thinking mode (e.g., Qwen3-VL Instruct, Qwen3.5, Gemma-4-IT), Phase 1 disables extended thinking (enable_thinking=False) to minimize output verbosity during summarization. Phase 2 enables it with a budget of 8,192 thinking tokens per question, allowing the model to reason over the full chain diary before committing to an answer.

J.4 Multi-Day Chain Handling

Each evaluation task specifies an ordered chain of video recordings spanning one or more simulated days, along with a stop_day_position and a global stop time in seconds marking the task's temporal scope. The pipeline assembles the diary text as follows: for days before the stop day, all four cached [Window summary] blocks are included. For the stop day, windows whose entire interval precedes the stop time are included in full; the window that straddles the stop time is re-summarized using only frames up to the stop time; and subsequent windows are excluded. This ensures the model is never exposed to events beyond the question's designated horizon.

J.5 Agentic Baselines

Adapting M3-Agent to SimLife-BP. M3-Agent (Long et al., 2026) handles long videos by separating memory construction from question answering. Instead of reasoning over the full video directly, it first converts the video into an explicit memory graph. To build this graph, M3-Agent splits the video into short clips, detects faces and speakers with existing tools, transcribes speech, and uses a fine-tuned Qwen2.5-Omni-7B model to generate structured descriptions for each clip. These descriptions use entity placeholders for people and voices, which are linked across clips to form a graph of multimodal memories. At test time, a fine-tuned Qwen3-32B model serves as the reasoning module: it repeatedly retrieves relevant memories from the graph using natural-language queries and answers questions based on the retrieved information.

In our SimLife adaptation, we do not train or fine-tune any M3-Agent models. Instead, we directly use the released M3-Agent models for inference and only modify the memory graph construction process. SimLife consists of reusable Sims video units, and each QA task combines multiple units into a longer chain. Processing every full chain from scratch would therefore be redundant. We first precompute clip-level memories for each reusable unit, including face clusters, speaker information, and Qwen2.5-Omni captions. For each QA task, we then assemble the memory graph by replaying the cached unit memories in the order specified by the task chain, mapping local entity placeholders to global identities, and constructing two graphs: one with audio information and one without. During evaluation, we select the appropriate graph according to whether audio is available in the task setting.

We make three lightweight SimLife-specific changes to improve reliability. First, instead of using ASR-generated transcripts, we use the clean dialogue text provided by SimLife when constructing textual memories. The input remains the original MP4 video, so the multimodal model can still perceive visual content and ambient audio, while the memory graph receives accurate transcript text. Second, we link the four main characters, Father Sim, Mother Sim, Daughter Sim, and Son Sim, to fixed global face identities using reference avatar embeddings. This improves identity consistency across reused video units. Third, we add the date and weekday from the chain calendar to each memory and store them as metadata, enabling more reliable retrieval for time-related questions.

Adapting VideoTree to SimLife-BP. VideoTree (Wang et al., 2025b) performs long video understanding by building a query-adaptive, hierarchical video representation to aid LLM reasoning in question answering. Rather than uniformly sampling frames, it performs adaptive breadth expansion to identify representative keyframes and then uses relevance-guided depth expansion to further explore the hierarchical structure to yield a tree-based representation of the video where the leaves are a question-focused subset of frames.

In our SimLife adaptation, we precompute frame-level EVA-CLIP-8B features once per reusable video unit and cache them on disk. When constructing a task, we assemble the full feature tensor by concatenating cached unit features in the order specified by the task chain, rather than re-encoding the same video multiple times. Caption resolution is also lazy: instead of loading all captions upfront, we store a narration manifest per chain that records each unit's caption path, global time offset, and calendar metadata (day index, day of week, and date). During tree construction and QA, only the captions for the sparse set of selected tree frames are loaded from disk.

We make a few SimLife-specific changes to the prompts given to the LLM at the breadth-expansion stage and the final question answering stage. First, we inject day and wall-clock time annotations into the narration: each caption line is prefixed with its simulated time (mapped from video-local seconds to the 06:00–22:00 simulation day) and grouped under a day header showing the date and weekday. This provides the LLM with explicit temporal grounding for time-sensitive questions.

For tasks in the audio-available setting, we augment the breadth-expansion prompt with dialogue, inserting speech events that fall within a five-second window of each sampled frame. This allows the model to incorporate conversational context when scoring frame relevance, without modifying the tree-construction algorithm itself. Finally, VideoTree's original prompts are designed for five-option multiple-choice questions, so we use **SIMLIFE**-specific prompt templates that format questions with two options (A and B) while preserving the relevance-scoring mechanism needed for breadth expansion.

K TEXT-ONLY BASELINE IMPLEMENTATION DETAILS

We evaluate two families of text-only baselines that receive no video frames — only the structured activity logs extracted during post-processing. These baselines serve two distinct roles: Text-Only (Noisy) measures how much can be inferred from the raw, automatically extracted log stream, and Text-Only (Noiseless) provides an oracle upper bound by supplying only the task-relevant subset of that stream. Both families are evaluated matching the hint variants used in all other baselines to enable direct comparison.

We use vllm 0.19.0 with 1~4 Nvidia A40 GPUs for inference task. It takes between several hours to 7 days to fully run the evaluation for one model.

K.1 INPUT REPRESENTATION

Log format. Activity logs are pre-materialized as plain-text files, one per simulated day. Each line encodes either a character action or a dialogue utterance with its in-game timestamp in HHMM format:

```
Action: HHMM Participant1, Participant2: [(category)] action - target
Dialogue: HHMM Speaker (to Others): "utterance"
```

Lines within a day are sorted first by in-game timestamp, then by original insertion order to break ties. Day sections are delimited by a header of the form `=== Day N (day_of_week, date) ===`, allowing the model to track the calendar structure of the simulation.

Noisy baseline concatenation. For each task, per-day log files are concatenated in chronological order up to and including the day designated by `stop_day_position`. Two log variants are used depending on the task type: an actions-only file (omitting dialogue) for standard tasks, and a file containing both actions and overlay-side dialogue for tasks that require audio–visual reasoning. For the final day in a chain, if the task specifies a mid-video stopping point (`stop_video_time_local`), the last day's text is rebuilt from the underlying log file by filtering events whose video timestamp precedes the cut.

When the resulting prompt exceeds the model's input budget, the earliest days are dropped until the truncated log fits. The retained suffix is the most temporally proximate context, which is generally most relevant to questions about behavior up to `stop_day_position`.

Noiseless baseline. The noiseless variant loads pre-materialized, oracle-filtered log files. Each file contains three sections — one per hint variant — and each section retains only: (i) day-boundary headers, (ii) the condition-side rows that the phenomenon predicate depends on, and (iii) occurrences of the target activity on fully observed prior days.

K.2 PROMPT STRUCTURE

Both baselines use an identical two-turn prompt structure (system + user). The system message instructs the model to answer only from the provided logs, to attend to day boundaries and day-of-week patterns, and to emit a strictly formatted response:

```
Your entire response MUST consist of exactly two lines:
Line 1:  Thought:  <at most 4 short sentences of reasoning>
Line 2:  Answer:  <a single capital letter A, B, C, or D>
```

The noiseless system prompt additionally notes that the input is a condensed observation history produced by oracle filtering, and that information beyond the observation cutoff should not be assumed.

The user message presents the concatenated log (or condensed history), followed by the question and lettered answer options.

K.3 Retry Policy

Each variant call allows up to three attempts. Attempts beyond the first append a tightening hint to the system message, which: (i) reminds the model that its previous attempt exhausted the token budget without emitting an answer, (ii) caps the Thought to three sentences, (iii) forbids nested counterfactual reasoning and day-by-day re-enumeration — the specific failure modes observed during development on reasoning-capable models. Retry delays are 0 s, 5 s, and 15 s for attempts 1, 2, and 3 respectively. The final attempt's output is persisted regardless of success; an unparseable response is scored as incorrect.