EGOROUTINE-BENCH: DISCOVERING RECURRING BEHAVIORAL PATTERNS FROM LONG-TERM EGOCENTRIC MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce EgoRoutine-Bench, a benchmark for discovering routines from long-term egocentric memory by jointly identifying recurring actions and their recurrence conditions. Existing long-horizon egocentric benchmarks largely evaluate behavioral reasoning through queries that specify the target behavior, leaving open whether models can discover which routines exist in a long-term history and under what conditions they recur. EgoRoutine-Bench addresses this gap by evaluating the joint discovery of routine actions and their recurrence conditions without predefined target actions. To make this directly evaluable, we explicitly define routines as action–recurrence-condition pairs and use a controlled semi-synthetic construction to inject corresponding synthetic event descriptions into natural egocentric caption histories. The benchmark further includes a multiple-choice next-action prediction task that tests whether discovered routine memory is useful for anticipating future behavior in synthetic caption streams. We also propose Routine Memory Induction (RMI) as a structured baseline that discovers candidate actions, verifies their occurrences, and consolidates contextual and temporal evidence to induce recurrence conditions and construct reusable routine memory. On EgoRoutine-Bench, full-history prompting and existing memory agents largely fail at joint routine discovery with Qwen3.5-2B and Qwen3.5-4B, achieving joint F1 scores no higher than 2.16. RMI provides a stronger starting point, achieving 7.1–14.2$\times$ the joint discovery F1 and improving next-action prediction accuracy by 6.12–14.29 percentage points over the strongest same-backbone memory-agent baselines.

1 INTRODUCTION

Resource-efficient on-device AI offers a path toward wearable assistants that understand and support users through their everyday experience (Huang et al., 2025; Kim et al., 2026) while keeping personal histories local. By continuously observing life from a first-person perspective, egocentric streams provide a record not only of what users see, but also of the actions they take and the contexts in which those actions occur (Grauman et al., 2022; Chen et al., 2024; Ye et al., 2025; Zhang et al., 2026). Beyond retaining past experiences in memory, these records provide evidence for anticipating future behavior. Research on habitual behavior shows that recurring contexts can cue actions repeatedly performed in those contexts (Neal et al., 2012). A user's history can therefore help an assistant anticipate what the user is likely to do when similar conditions arise again. Such anticipation can help the assistant decide whether support would be useful, what to offer, and when to offer it, without waiting for an explicit request. To ground this form of proactive assistance in a user's history, the assistant must identify the user's routines: *which actions recur and under what contextual conditions*.

Existing long-horizon egocentric benchmarks test behavioral reasoning over extended histories (Yang et al., 2025; Wang et al., 2026; Chen et al., 2026), but largely through questions that specify which behavior to examine. For example, "When do I usually check the shuttle time after packing my laptop?" specifies the target action before the system retrieves or reasons over its history (Figure 1, top). These evaluations leave open whether models can discover recurring actions and their recurrence conditions without any prior specification of the target actions, and whether the discovered routines can help predict the user's next action in new contexts.

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.

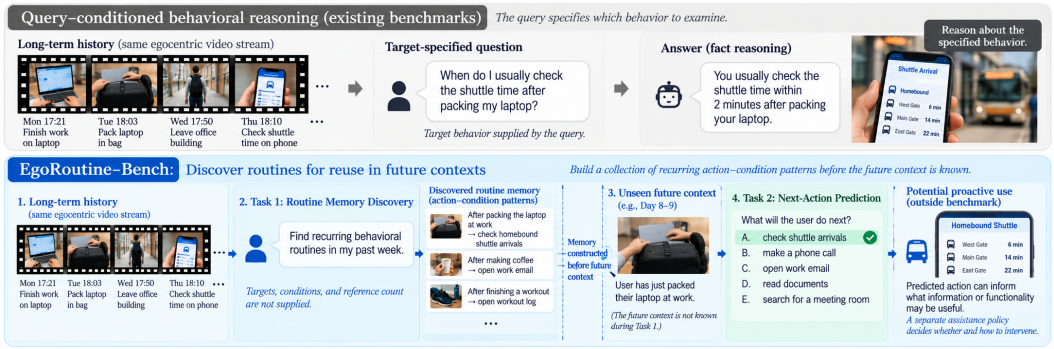


Figure 1: **EgoRoutine-Bench: discovering routines to anticipate future actions.** Unlike query-conditioned reasoning about a specified behavior (top), routine discovery identifies recurring actions and their recurrence conditions without either being supplied (bottom). The discovered memory is reused in a multiple-choice task for next-action prediction.

To address this problem, we introduce EgoRoutine-Bench, a text-only benchmark that evaluates routine discovery from long-term egocentric history (Figure 1, bottom). Given a multi-day egocentric caption history, the system must recover a collection of recurring actions and their recurrence conditions without being given either. To make this discovery problem directly measurable, we construct controlled semi-synthetic histories from EgoLife captions (Yang et al., 2025). We inject synthetic event descriptions into natural caption histories according to predefined routines, replacing selected caption segments with paraphrased descriptions of those events while retaining the surrounding everyday activity. For each predefined routine, we retain its action, recurrence condition, and the exact injected events generated from that routine as ground truth for evaluating joint action–condition discovery. EgoRoutine-Bench further includes a multiple-choice question (MCQ) task for next-action prediction to test whether routine memory discovered from historical captions can be reused to anticipate future behavior.

We also introduce Routine Memory Induction (RMI) as a structured baseline for constructing routine memory from long-term egocentric history. RMI discovers candidate actions, retrieves and verifies their occurrences, and consolidates contextual and temporal evidence to induce recurrence conditions. The resulting action–condition pairs form a compact routine memory that can be reused for future prediction. RMI provides a stronger starting point than the existing memory-agent baselines, improving both joint routine discovery and given-action condition induction. Its discovered routine memory also improves downstream next-action prediction over full-history prompting and the evaluated memory-agent inputs.

Our contributions are summarized as follows.

- We introduce EgoRoutine-Bench, a controlled semi-synthetic benchmark for joint action–condition discovery from long-term egocentric memory without supplied target actions or recurrence conditions, together with a multiple-choice next-action prediction task that evaluates the downstream utility of discovered routine memory.
- We introduce Routine Memory Induction (RMI), a structured baseline for routine discovery that identifies candidate actions, verifies their occurrences, and induces recurrence conditions to construct reusable routine memory.
- We find that RMI provides a stronger baseline than the existing long-term memory agents for joint routine discovery, while its discovered routine memory also improves next-action prediction over full-history prompting and the evaluated memory-agent inputs.

2 Related Work

2.1 Egocentric Lifelogging and Behavioral Benchmarks

Egocentric lifelogging has expanded from activity understanding to memory and behavioral QA. EPIC-KITCHENS (Damen et al., 2018) and Ego4D (Grauman et al., 2022) support action understanding and anticipation, while EgoSchema (Mangalam et al., 2023) and HD-EPIC (Perrett et al.,

2025) evaluate video QA. SuperMemory-VQA (Alam et al., 2026) tests long-horizon visual recall, while LifeDialBench (Zheng et al., 2026) evaluates memory through simulated lifelog dialogues. EgoLifeQA (Yang et al., 2025) evaluates long-context question answering for everyday assistance, EgoMemReason (Wang et al., 2026) tests reasoning over entity, event, and behavior memories across days, and EgoMonth (Chen et al., 2026) evaluates long-term spatiotemporal memory over month-scale egocentric recordings. Relatedly, Menchón et al. (2020) discover behavioral routines by clustering semantically similar time segments in egocentric photo-streams. In contrast to both such clustering-based discovery and action-specified behavioral QA, our setting requires recovering explicit action–condition patterns from multi-day histories without supplied target actions.

2.2 Long-Term Memory Agents

LLM long-term memory systems organize textual histories through tiered storage in MemGPT (Packer et al., 2023), linked notes in A-Mem (Xu et al., 2025), community summaries in GraphRAG (Edge et al., 2024), and temporal knowledge graphs in Zep (Rasmussen et al., 2025). Multimodal systems extend memory retrieval to video and audio. AMEGO (Goletto et al., 2024) clusters object-interaction tracks and locations, while VideoRAG (Ren et al., 2026) combines caption/transcript graphs with visual retrieval. EgoGraph (Sun et al., 2026) and AVA (Yan et al., 2026) organize entities and events in temporal graphs. WorldMM (Yeo et al., 2026) and M3-Agent (Long et al., 2026) combine episodic and semantic memories with iterative query generation and retrieval. EgoRAG (Yang et al., 2025) localizes time windows before retrieving detailed captions, while EgoMemo (Sitong et al., 2026) maintains streaming temporal, semantic, and visual memories with hierarchical caption retrieval through selected hourly windows. These systems primarily retrieve and synthesize historical evidence conditioned on a supplied query or current context, rather than explicitly discovering recurring actions and their recurrence conditions for persistent storage. We address this modeling gap with a framework that constructs reusable action–condition abstractions directly from long-term history.

3 EgoRoutine-Bench

EgoRoutine-Bench evaluates whether models can discover recurring actions and their recurrence conditions from long-term egocentric caption histories without either being supplied. We score discovered routines against predefined ground truth and test whether the resulting routine memory improves next-action prediction in an MCQ task. Figure 2 illustrates the construction pipeline.

3.1 Problem Formulation

To distinguish routines from actions that merely repeat, we define a *routine* as an action that recurs under a consistent contextual or temporal condition in the observed history. We represent each routine as a pair $r = (a, \rho)$, where a is the action and ρ is its recurrence condition. A recurrence condition summarizes how an action recurs in the observed history; it does not imply that the action occurs whenever the condition holds.

Joint discovery. Let $\mathcal{M}_{\leq T} = \big((x_i, \tau_i)\big)_{i=1}^{N}$ denote a history of N captions available up to cutoff T, where x_i is a caption and τ_i is its timestamp or temporal interval. A routine discovery system D takes this history and a general discovery instruction ι as input and returns a collection of inferred routines:

$$\widehat{\mathcal{R}} = D(\mathcal{M}_{\leq T}, \iota) = \big((\hat{a}_k, \hat{\rho}_k)\big)_{k=1}^{\hat{K}}, \tag{1}$$

where $\hat{K}$ is the number of routines returned by the system. The instruction provides routine types, their definitions, and few-shot examples, but does not specify the target actions, their recurrence conditions, or the number of routines to discover. The system may generate its own queries and retrieve evidence from the history.

Next-action prediction. To evaluate whether discovered routines help predict later actions, we use $\widehat{\mathcal{R}}$ as routine memory in a multiple-choice question (MCQ) task. Each question is constructed around a target occurrence of a ground-truth routine at time $t_j > T$; c_j is its pre-target caption prefix and $\mathcal{C}_j$ the set of five candidate actions. A next-action predictor P takes this context, the candidate actions,

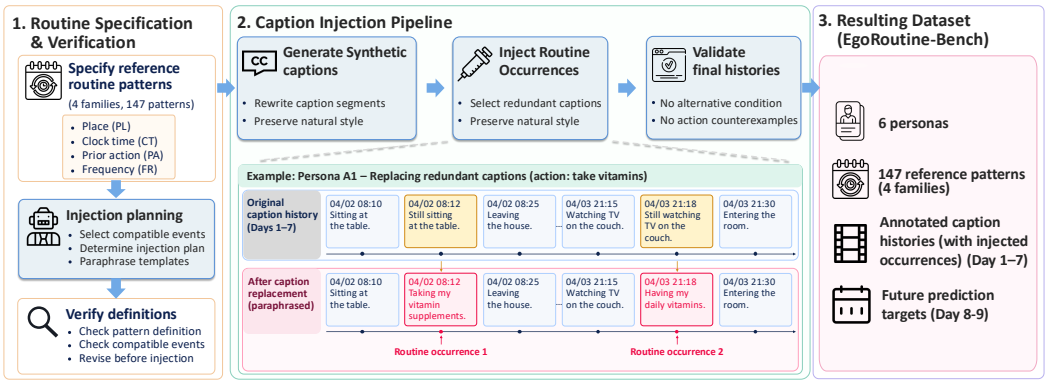


Figure 2: **Construction and annotation pipeline of EgoRoutine-Bench.** We inject routines into EgoLife histories following their recurrence conditions and verify injected and background occurrences in Days 1–7. Days 8–9 are synthesized separately for next-action prediction.

Table 1: **Recurrence-condition families in EgoRoutine-Bench.** Examples are drawn from the benchmark's annotated routines; construction details appear in Appendix A.1.

Family	Condition	Example
Place (PL)	Action repeats at a particular place	Log caffeine intake at the desk.
Clock time (CT)	Action repeats at a particular time	Check the budgeting app around 11:00.
Prior action (PA)	Action repeats after another action	Open the blinds after pouring tea.
Frequency (FR)	Elapsed interval between successive occurrences, or count per stated period	Take a throat lozenge every 2 hours; apply acne ointment twice a day.

and the discovered routine memory as input and selects an action:

$$\hat{a}_j = P(c_j, \mathcal{C}_j, \widehat{\mathcal{R}}) \in \mathcal{C}_j. \tag{2}$$

In EgoRoutine-Bench, T is the end of Day 7, and prediction targets occur on Days 8–9. The routine memory is constructed from Days 1–7 and held fixed; Appendix C.3 details target selection and prefix construction.

3.2 Data Generation and Annotation Pipeline

We construct semi-synthetic histories for routine discovery by replacing selected EgoLife captions (Yang et al., 2025) with synthetic descriptions of routine occurrences. We annotate and verify these Days 1–7 histories, then separately synthesize Days 8–9 captions for next-action prediction. Figure 2 summarizes the construction and annotation pipeline. We use Claude Opus-5 for routine specification, synthetic caption generation, and reference verification.

Routine specification. To construct ground-truth routines, we first define four recurrence-condition types, as illustrated in Table 1. *Place* (PL), *clock time* (CT), and *prior action* (PA) specify recurrence at a particular location, at a particular time of day, and after a particular action, respectively. These conditions are motivated by research on place cues in habitual behavior (Neal et al., 2012) and habit formation around specified times or existing activities (Keller et al., 2021). *Frequency* (FR) specifies an elapsed interval between successive occurrences or an occurrence count within a prespecified period. Appendix A.1 details occurrence placement and temporal tolerance. For each persona, we create a new set of actions and plausible conditions for each type, then pair selected actions and conditions to construct ground-truth routines.

Historical caption construction. For each ground-truth routine, we generate three to five paraphrased synthetic caption variants, each describing an individual occurrence of the action rather than explicitly stating the underlying routine. We integrate these synthetic occurrences into background histories composed of timestamped captions generated from real EgoLife videos following the EgoLife captioning pipeline (Yang et al., 2025). To realize each routine's specified recurrence condition,

we first determine the timestamps at which its occurrences should appear in the history. At each selected timestamp, we identify a nearby redundant background caption whose described activity is already represented by the surrounding captions, and replace it with one of the generated synthetic captions. This preserves the original temporal structure and surrounding activities: timestamps and all unselected captions remain unchanged. We use replacement rather than appending because appending adds text only at injected timestamps, creating a length-based artifact that makes the synthetic occurrences readily detectable.

Ground-truth verification. After constructing the histories, we verify each ground-truth routine against all occurrences of its action in the final Days 1–7 captions, including background occurrences. A routine is retained only if its specified condition remains consistent throughout the observation scope and no semantically distinct alternative condition for the same action also qualifies. Appendix A.1 provides the construction and reference-checking details.

Future caption synthesis. Because the real EgoLife recordings end at Day 7, evaluating next-action prediction requires extending each persona's history with future observations. We therefore synthesize Days 8–9 as continuations of the historical stream, preserving the same ground-truth routine definitions while instantiating new occurrences of those routines in future contexts. To form complete future caption streams, we interleave these new routine occurrences with synthetic background captions and describe them with new paraphrases rather than reusing the historical injected captions.

3.3 Dataset Validation and Statistics

Counterfactual evidence-dependence. We screen candidate references for evidence-dependent condition recovery by giving GPT-5.5 the target action and only its relevant occurrence captions from an original and a counterfactual setting. Each pair preserves the target action and occurrence count while changing the condition-defining evidence. We retain a reference only when the model correctly recovers the respective conditions in both settings. Appendix A.2 details this screening and its distinction from full-caption given-action evaluation.

Injection detectability. After injection, we test whether synthetic captions are readily identifiable and could therefore provide a shortcut for locating routine occurrences. We ask GPT-5.5 to classify each window as containing or not containing an injected caption, using windows of 12 consecutive captions from all personas, evenly split between windows with and without injections. We find that GPT-5.5 reports injections in 94% of windows with injected captions and 86% of those without, indicating that it fails to reliably distinguish the two groups in this local test (Appendix A.3).

The resulting benchmark spans six personas with seven days of history each (195.6 hours; 357,705 captions), containing 147 ground-truth routines supported by 764 annotated historical occurrences (mean 5.20 per routine; range 4–14). Appendix A.4 provides detailed dataset statistics, and Table 7 lists all ground-truth routines by recurrence-condition type.

4 Routine Memory Induction

We introduce Routine Memory Induction (RMI), a structured baseline for the routine discovery problem defined in Section 3.1. RMI first discovers candidate actions, then applies context extraction, evidence consolidation, and condition verification to infer the conditions under which they recur. The resulting pairs of actions and conditions form reusable routine memory for next-action prediction. The complete procedure is specified in Appendix B.

4.1 Candidate Action Discovery

Because target actions are not supplied, RMI first constructs a candidate inventory from the wearer's observed behavior.

Action extraction and grouping. An LLM extracts atomic descriptions of the wearer's actions from $\mathcal{M}_{\leq T}$, retaining source identifiers and timestamps. To group observations of the same action, we

embed these descriptions with Qwen3-Embedding-4B and construct a graph connecting observations only when they select each other as neighbors. The connected components form initial groups. To focus on recurring rather than one-off actions, we deduplicate temporally overlapping observations and retain only groups supported by at least two distinct occurrences.

Semantic group merging. Because the same action may remain split across groups, we first combine groups with identical canonical action labels, then propose further merges based on centroid cosine similarity. We accept these proposals only when an LLM judges the groups to describe the same action, yielding the candidate inventory $\mathcal{A}$ for pattern induction. Parameter settings are fixed across experiments and detailed in Appendix B.1.

4.2 Pattern Induction Pipeline

For each candidate action $a \in \mathcal{A}$, RMI retrieves relevant memories and processes them through the following three stages.

Context extraction: tagging, target-action filtering, and sorting. We query a dense index of atomic captions in $\mathcal{M}_{\leq T}$ using the candidate action and its action-only aliases. A caption-level target-action filter retains hits judged to describe the wearer performing the action. For each hit, we restore the surrounding captions; a context extractor then returns structured JSON containing action presence, place, an ordered action sequence with the target marked, and local state. We retain only records with positive action presence in this restored context and sort them by source day and start time, yielding $\mathcal{P}_a$.

Evidence consolidation: canonicalization and overlap merging. An action-specific map ν_a canonicalizes equivalent place and contextual-action expressions separately, choosing representative labels from $\mathcal{P}_a$ while preserving source identity and temporal order. We then merge each connected component of overlapping windows on the same day into one record to obtain chronological occurrence evidence $\mathcal{E}_a$.

Condition verification. The condition verifier is invoked only when at least two records remain *after* overlap merging, i.e., $|\mathcal{E}_a| \geq 2$; otherwise, the procedure returns `None`. For eligible actions, we precompute consecutive temporal gaps from $\mathcal{E}_a$. The verifier uses these gaps and consolidated context to infer and assess a condition under four prompt-defined pattern types: clock time, periodicity, place, and adjacent action. It returns a supported condition or `None`; only non-null action–condition pairs enter routine memory. Appendices B.2–B.4 specify the three stages.

5 Experiments

We evaluate whether systems can discover an action–condition routine collection without supplied target actions and whether the resulting memory supports subsequent next-action prediction.

5.1 Experimental Setup

Task setup. For joint discovery, systems receive Days 1–7 captions as historical memory, definitions of the routine types, and few-shot examples; target actions and the number of routines to discover are not specified. For next-action prediction, each question targets a ground-truth routine occurrence in Days 8–9; systems receive its pre-target caption prefix and select the next action from five candidates.

Baselines. With on-device assistance as our motivating setting, we compare RMI with *Full captions* and memory agents using Qwen3.5-2B and Qwen3.5-4B. Discovery baselines use *Direct*, which outputs actions and conditions together, or *Action-first*, which enumerates actions before inferring their conditions. Next-action prediction compares full captions, agents' retrieved evidence (*Ret.*), and routine memory from their *Action-first* outputs (*Disc.*). Appendix C.1 details the baselines and resource settings; Appendix C.3 specifies the prediction inputs.

Table 2: **Joint action–condition discovery without supplied target actions.** Values are averaged over three runs. Scores use GPT-5.6-Luna judgments. Action evaluates action matching; Joint also requires matching recurrence conditions. Both report precision (P), recall (R), and F1. Discovered Routines counts predictions excluding semantic matches to the neutral list.

Backbone	Method	Discovery protocol	Discovered Routines	Action			Joint		
				P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑
With full captions									
Qwen3.5-4B	Full captions	Direct	4.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	8.00	25.26	1.36	2.58	0.00	0.00	0.00
Claude Sonnet-5	Full captions	Direct	50.33	53.19	17.92	26.74	18.72	6.35	9.46
		Action-first	72.67	58.19	28.57	38.30	19.76	9.75	13.05
Gemini-3.5-Flash	Full captions	Direct	30.00	87.96	17.92	29.75	71.46	14.52	24.11
		Action-first	57.00	85.42	32.88	47.44	43.78	16.78	24.24
GPT-5.5	Full captions	Direct	51.33	92.05	31.97	47.35	46.31	15.88	23.59
		Action-first	136.67	83.54	77.55	**80.42**	46.85	43.54	**45.12**
Memory agents									
Qwen3.5-4B	EgoRAG (Yang et al., 2025)	Direct	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	105.33	3.81	2.72	3.17	0.95	0.68	0.79
	Ego-R1 (Tian et al., 2026)	Direct	12.33	16.24	1.36	2.51	8.12	0.68	1.26
		Action-first	38.00	35.09	9.07	14.41	5.27	1.36	2.16
	WorldMM (Yeo et al., 2026)	Direct	11.67	17.25	1.36	2.52	0.00	0.00	0.00
		Action-first	10910.67	0.93	68.93	1.84	0.51	37.64	1.00
	EgoMemo (Sitong et al., 2026)	Direct	330.33	0.30	0.68	0.42	0.30	0.68	0.42
		Action-first	3124.67	0.21	4.53	0.41	0.18	3.85	0.35
	M3-Agent (Long et al., 2026)	Direct	168.00	2.40	2.72	2.54	0.60	0.68	0.64
		Action-first	667.33	1.65	7.48	2.71	0.60	2.72	0.99
	RMI (ours)	Action-first	209.00	29.04	41.05	**33.98**	26.23	37.19	**30.73**

Table 3: **Next-action prediction (five-choice accuracy, %; 3-run average).** History input supplements the current context and question: none (None), full captions (Full), retrieved evidence (Ret.), discovered routines (Disc.), or ground-truth routines (Oracle). None and Full use no memory agent.

Answer model	Oracle	None	Full	EgoRAG		Ego-R1		WorldMM		EgoMemo		M3-Agent		**RMI**
				Ret.	Disc.	Ret.	Disc.	Ret.	Disc.	Ret.	Disc.	Ret.	Disc.	Disc.
Qwen3.5-4B	94.78	23.13	40.59	39.46	36.73	31.52	23.81	32.65	44.22	39.23	36.05	36.05	29.25	**50.34**
Qwen3.5-2B	93.20	20.41	21.09	20.41	14.97	20.41	12.93	23.13	27.21	20.41	17.69	17.69	22.45	**41.50**

Evaluation metrics. Discovery reports micro-averaged precision (P), recall (R), and F1 for *Action* matching (action alone) and *Joint* matching (action and recurrence condition of the same ground-truth routine), with joint F1 as the primary metric. For each metric, only the highest-scoring eligible prediction per reference receives credit; further matches are duplicate false positives. Within each persona, distinct ground-truth actions have unique reference conditions, so each predicted pair can match at most one reference. Matches to a precompiled background inventory are neutral (neither credited nor penalized); this excludes actions semantically equivalent to any ground-truth action, making ground-truth and neutral matches mutually exclusive. Next-action prediction reports accuracy across five-choice questions over all ground-truth routines. Appendix C.2 details the protocol.

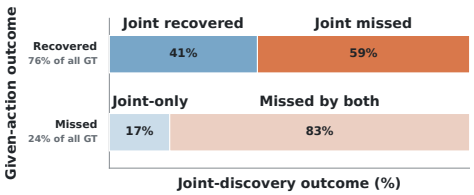
5.2 Main Results

Joint Action–Condition Discovery Table 2 reports the 4B results; Qwen3.5-2B results appear in Appendix Table 8. Full-caption prompting with compact backbones recovers no complete action–condition pairs, while memory agents exhibit uneven trade-offs between precision and recall. RMI reaches joint F1 scores of 30.73 with 4B and 12.41 with 2B, compared with 2.16 and 1.76 for the strongest same-backbone memory-agent baselines. Its advantage combines routine coverage with more selective recovery of routine-related actions and their conditions. For example, RMI-4B and WorldMM's *Action-first* variant have similar joint recall (37.19 versus 37.64), but RMI has substantially higher joint precision (26.23 versus 0.51). With 2B, RMI exceeds all same-backbone

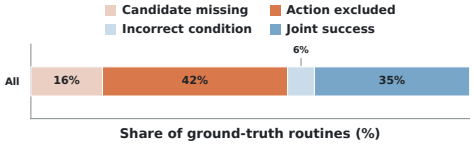
Table 4: **Given-action condition induction.** All entries report accuracy averaged over three evaluations by GPT-5.6-Luna. The first block reports full-caption baselines; the second compares memory-agent baselines and RMI with Qwen3.5-2B and Qwen3.5-4B.

Full captions					
Metric	GPT-5.5	Sonnet-5	Gemini-3.5-Flash	Qwen3.5-4B	Qwen3.5-2B
Acc. (%) ↑	57.37	31.75	38.78	10.20	3.40

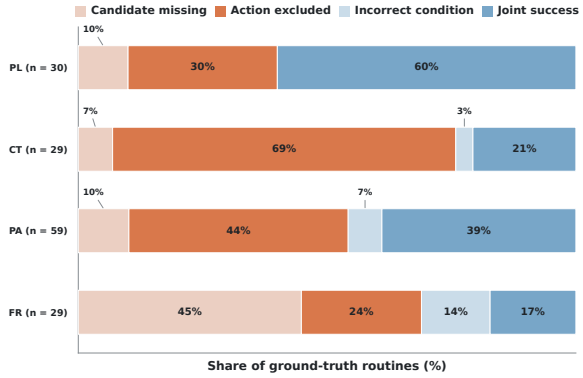
Memory agents						
Backbone	EgoRAG	Ego-R1	EgoMemo	WorldMM	M3-Agent	**RMI (ours)**
Qwen3.5-4B	13.15	12.93	33.11	55.33	42.86	**76.42**
Qwen3.5-2B	12.93	12.24	14.29	42.63	30.61	**61.90**



(a) Given-action vs. joint discovery



(b) Stage-wise failure analysis



(c) Failure analysis by recurrence type

Figure 3: **Routine-level discovery gap and failure analysis for RMI-4B.** Results are from one of the three random seeds. (a) Joint-discovery outcomes conditional on the given-action outcome, using routine-ID-paired observations and normalized within each given-action group. (b) Outcomes of all ground-truth routines, distinguishing missing candidate actions, actions excluded from the final routine output, incorrect recurrence conditions, and successful joint recovery. (c) The same outcome breakdown within each recurrence-condition type.

memory-agent baselines in both joint precision and recall. However, RMI-4B remains below full-caption GPT-5.5 with *Action-first* (45.12 joint F1), and many retained predictions still fail to match reference routines.

Next-Action Prediction Table 3 compares next-action prediction using full captions, retrieved evidence, and discovered routines. Retrieval uses the current context and prediction question, without a query specifying the target action. We first observe that ground-truth routines enable high prediction accuracy even with the 2B model, whereas the current context alone yields near-chance accuracy. Yet full captions and retrieved evidence provide little benefit to 2B, and even 4B remains far below the oracle with these inputs. RMI's discovered routine memory reaches 50.34% accuracy with 4B and 41.50% with 2B, compared with 40.59% and 21.09% for full captions. It exceeds the strongest memory-agent inputs, WorldMM's discovered routines (44.22% and 27.21%), by 6.12 and 14.29 percentage points, respectively. These gains show that RMI's routine collection provides useful action–condition memory for next-action prediction, although accuracy remains well below the oracle.

5.3 Analysis and Ablation

Table 2 shows that recovering actions does not ensure recovering their conditions: full-caption GPT-5.5 with *Action-first* reaches 80.42 action F1 but 45.12 joint F1. RMI-4B has a smaller action–joint F1 gap (33.98 versus 30.73) and higher joint precision than same-backbone baselines, while its joint recall of 37.19% leaves substantial room to improve coverage. We examine how candidate actions become retained routine pairs through given-action evaluation, paired routine outcomes, and pipeline traces.

Table 5: **RMI component analysis.** Accuracy (%) after cumulatively adding each component.

Component	4B	2B	0.8B
Retrieval only	58.50	42.18	12.24
+ Context extraction	73.02	59.86	22.90
+ Evidence consolidation + Condition verification	**76.42**	**61.90**	**32.65**

Table 6: **Ablation results of memory variants on next-action prediction.**

Memory input	Acc.
No Memory	23.13
Action-only	36.73
Shuffled	32.43
RMI	**50.34**

RMI remains strong when the action is supplied. We isolate condition induction from action discovery by supplying each ground-truth action and asking methods to recover its recurrence condition from the same history. Results in Table 4, averaged over three evaluations by GPT-5.6-Luna, show that RMI reaches 76.42% with 4B and 61.90% with 2B. These scores exceed the strongest same-backbone memory-agent baseline, WorldMM (55.33% and 42.63%), and full-caption GPT-5.5 (57.37%). Appendix D.2 reports the individual judge scores.

Given-action success does not ensure joint discovery. We next examine whether routines recovered when their actions are supplied are also recovered when the target action must be discovered. Figure 3(a) reports RMI-4B outcomes for one of the three random seeds, separately for routines recovered and missed in the given-action setting. For this seed, joint discovery misses 58.9% of the routines successfully recovered in the given-action setting, recovering only 41.1% of this group. Thus, given-action evaluation alone does not measure the failures that arise when the target action must also be discovered.

Most routine losses occur after candidate discovery. We further trace each ground-truth routine through RMI-4B's candidate inventory and final routine output. Figure 3(b) shows that, among the 95 routines not jointly recovered, only 24 lack a matching candidate action, whereas 71 fail after a matching candidate has already been identified. Of these, 62 are present in the candidate inventory but absent from the final routine output, and 9 retain the action with an incorrect condition. Thus, 74.7% of joint-discovery failures in this seed occur after candidate discovery, highlighting the need to retain more correct action–condition pairs while controlling false positives. Figure 3(c) further shows that the dominant failure stage varies by recurrence type: clock-time routines are most often excluded from the final output after candidate discovery (69.0%), whereas frequency routines most often lack a matching candidate action (44.8%).

Component analysis of RMI. Table 5 adds pattern-induction components cumulatively to dense retrieval. Context extraction improves accuracy across all three backbones. Adding evidence consolidation and condition verification together raises accuracy to 76.42%, 61.90%, and 32.65% for 4B, 2B, and 0.8B, respectively, with the largest additional gain on 0.8B (+9.75 percentage points).

Prediction-memory intervention. We test the contribution of conditions and their pairing with actions. RMI retains all discovered action–condition pairs; *Action-only* removes conditions while preserving the action list; and *Shuffled* preserves the condition multiset but reassigns eligible links within each persona and predicted routine type. Shuffled results are averaged over three seeds. In Table 6, RMI reaches 50.34% accuracy, compared with 23.13% without memory, 36.73% with Action-only, and 32.43% with Shuffled. Its gains of 13.61 points over Action-only and 17.91 points over Shuffled support the value of both condition information and its correct pairing with actions.

6 Conclusion

We introduced EgoRoutine-Bench, a controlled semi-synthetic benchmark for discovering recurring actions and their conditions from long-term egocentric caption histories without supplied target actions or recurrence conditions. We also proposed RMI, which constructs reusable routine memory by discovering candidate actions, verifying their occurrences, and inducing recurrence conditions. In the reported experiments, RMI improves joint routine discovery and given-action condition induction over evaluated memory-agent baselines, while its discovered memory improves next-action prediction over full-history prompting and the evaluated memory-agent inputs.

AI USE STATEMENT

We conceived the research, developed the methodology, designed the experiments, and interpreted the results. We used generative models for benchmark construction, validation, and model-based evaluation, as detailed in the paper. AI tools also assisted with baseline-reproduction setup under our specified configurations, debugging, execution-script preparation, English proofreading, and figure production (Figure 1, generated using Codex). We manually reviewed all synthetic data, inspected AI-assisted code before execution, and verified the reported numerical results and cited literature. We reviewed all AI-assisted text and figures and take full responsibility for the final manuscript, claims, and artifacts.

REFERENCES

Samiul Alam, Shakhrul Iman Siam, Michael J Proulx, James Fort, Richard Newcombe, Hyo Jin Kim, and Mi Zhang. Supermemory-vqa: An egocentric visual question-answering benchmark for long-horizon memory. *arXiv preprint arXiv:2606.00825*, 2026.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *CVPR*, 2024.

Weitao Chen, Hu Jiaxin, Xie Tianyidan, Yang Li, Yuyi Qian, Banghao Xu, Ziheng Tang, Shenyi Wang, Mingyue Yu, Duo Li, et al. Egomonth: A month-level egocentric video benchmark for long-term spatiotemporal memory. *arXiv preprint arXiv:2608.13113*, 2026.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: the dataset. In *ECCV*, 2018.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *ECCV*, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

Yifei Huang, Jilan Xu, Baoqi Pei, Lijin Yang, Mingfang Zhang, Yuping He, Guo Chen, Xinyuan Chen, Yaohui Wang, Zheng Nie, et al. Vinci: A real-time smart assistant based on egocentric vision-language model for portable devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–33, 2025.

Jan Keller, Dominika Kwasnicka, Patrick Klaiber, Lena Sichert, Phillippa Lally, and Lena Fleig. Habit formation following routine-based versus time-based cue planning: A randomized controlled trial. *British journal of health psychology*, 26(3):807–824, 2021.

Yoonsang Kim, Devshree Jadeja, Divyansh Pradhan, Yalong Yang, and Arie E Kaufman. SpeechLess: Micro-utterance with personalized spatial memory-aware assistant in everyday augmented reality. In *IEEE VR*, 2026.

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 *ICLR*, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

Martín Menchón, Estefanía Talavera, José Massa, and Petia Radeva. Behavioural pattern discovery from collections of egocentric photo-streams. In *ECCV*, 2020.

David T Neal, Wendy Wood, Jennifer S Labrecque, and Phillippa Lally. How do habits guide behavior? Perceived and actual triggers of habits in daily life. *Journal of experimental social psychology*, 48(2):492–498, 2012.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, et al. Hd-epic: A highly-detailed egocentric video dataset. In *CVPR*, 2025.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. In *KDD*, 2026.

Gong Sitong, Tianyu Yan, Caixin Kang, Bo Zheng, Xiang Ruan, Huchuan Lu, Kaipeng Zhang, Yoichi Sato, and Yifei Huang. Vinci2: Providing proactive assistance in continuous egocentric videos. In *ECCV*, 2026.

Shitong Sun, Ke Han, Yukai Huang, Weitong Cai, and Jifei Song. EgoGraph: Temporal knowledge graph for egocentric video understanding. *arXiv preprint arXiv:2602.23709*, 2026.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-R1: Agentic chain-of-tool-thought for ultra-long egocentric video reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Ziyang Wang, Yue Zhang, Shoubin Yu, Ce Zhang, Zengqi Zhao, Jaehong Yoon, Hyunji Lee, Gedas Bertasius, and Mohit Bansal. EgoMemReason: A memory-driven reasoning benchmark for long-horizon egocentric video understanding. In *COLM*, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *NeurIPS*, 2025.

Yuxuan Yan, Shiqi Jiang, Ting Cao, Yifan Yang, Qianqian Yang, Yuanchao Shu, Yuqing Yang, and Lili Qiu. AVA: Towards agentic video analytics with vision language models. In *NSDI*, 2026.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *CVPR*, 2025.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, et al. Mmego: Towards building egocentric multimodal llms for video qa. In *ICLR*, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *CVPR*, 2026.

Haoyu Zhang, Qiaohui Chu, Meng Liu, Haoxiang Shi, Yaowei Wang, and Liqiang Nie. Exo2ego: Exocentric knowledge guided mllm for egocentric video understanding. In *AAAI*, 2026.

Jianjie Zheng, Zhichen Liu, Zhanyu Shen, Jingxiang Qu, Guanhua Chen, Yile Wang, Yang Xu, Yang Liu, and Sijie Cheng. Evaluating memory capability in continuous lifelog scenario. In *Findings of ACL*, 2026.

A Caption Construction and Reference Criteria

A.1 Reference Support and Condition Validity

Reference checks and occurrence placement. Ground-truth verification considers distinct occurrences of each target action in the final Days 1–7 caption history, including both injected and background occurrences. Multiple captions describing the same event count as one occurrence. The construction schedule and condition parameters are set before validation. The placement rules below specify where synthetic support is inserted; they do not by themselves establish uninterrupted video coverage of an entire day.

Place, clock-time, and prior-action routines. For each routine, we schedule five occurrences by randomly sampling their day assignments from Days 1–7 with replacement. Multiple distinct occurrences may therefore fall on the same day, so five occurrences need not span five distinct days. Validation examines the action's occurrences throughout the seven-day caption history. Place and clock-time occurrences must agree with the annotated location and clock time or time window, respectively (Table 7). For prior-action routines, the specified contextual action must precede each target occurrence.

Daily counts and scheduled days. For "once per day," we insert one occurrence on each of Days 1–7. For "twice a day," we insert two occurrences on each of Days 1–7. This produces fourteen injected occurrences across seven days for each of `J-PF05` and `L-PF05` in Table 7. The first occurrence each day is typically placed in the morning, leaving time for both occurrences within that day. The two occurrences need not be equally spaced. For "every other day," we insert one occurrence on alternating days, such as Days 1, 3, 5, and 7. These alternating-day schedules differ from "twice a day," which places two occurrences on every day. All daily-count and alternating-day conditions are validated against the recorded history over Days 1–7, including days without scheduled injections. The table reports injected support only; background occurrences are also considered during reference validation and are excluded from these tabulated counts. The injection-day schedule alone does not establish the total number of action occurrences on other recorded days.

Hourly intervals and temporal tolerance. EgoLife captures participants' waking hours rather than continuous 24-hour histories (Yang et al., 2025). For "every N hours," the observation window for each persona-day runs from that day's first recorded timestamp to its last. These boundaries come from the source history, independently of where routine occurrences are injected. Unrecorded overnight and early-morning periods before recording begins lie outside the window. We place hourly occurrences within these daily recording windows on days sampled from Days 1–7. Hourly routines need not be confined to a single day: the six-occurrence four-hour routines `T-PF05` and `S-PF05` use two days, with three occurrences on each day. Within each day's window, let $t_1 < \cdots < t_m$ be the timestamps of all successive distinct recorded occurrences of the target action, including both injected and background occurrences. We require

$$|(t_{i+1} - t_i) - N\,\text{hours}| \leq 5\,\text{minutes}, \qquad i = 1, \ldots, m-1. \tag{3}$$

Every consecutive pair within the same window is checked; a boundary between separately injected sequences does not reset the check. Intervals crossing from one day's recorded window to another are excluded because the intervening overnight period is unobserved. For example, three recorded occurrences four hours apart within one day's window yield two checked intervals; the gap to the next day's first occurrence is outside this within-day rule. This criterion concerns successive recorded occurrences and does not assume uninterrupted recording or establish the absence of unobserved events during recording breaks. One occurrence per arbitrary N-hour clock bin is insufficient, as is a retrieved subset that omits intervening occurrences from the full caption history.

A.2 Counterfactual Evidence-Dependence Validation

Screening input and task. For each candidate routine, GPT-5.5 receives the explicitly specified target action and only its relevant occurrence captions, with their condition-defining temporal and contextual information. The full persona history and unrelated surrounding captions are excluded. The model is asked to infer the recurrence condition using the same instructions and output format as

the full-caption given-action diagnostic. This screening tests whether the condition is recoverable from its evidence and whether the prediction changes when that evidence changes.

Counterfactual construction and acceptance. We start with an initial pool of 170 candidate routines. We construct an original and a counterfactual occurrence set supporting different conditions within the same routine family. Each pair preserves the target action, occurrence count, question, and output format. We change only the condition-defining evidence: clock times for CT, day/time schedules for FR, associated locations for PL, and the action or object state immediately preceding the target for PA. The two sets are submitted to GPT-5.5 in separate stateless requests. GPT-5.6-Luna compares each predicted condition with its reference under the criteria in Appendix C.2. A candidate is retained only when GPT-5.5 correctly recovers the original condition from the original captions and the altered condition from the counterfactual captions. Of the 170 candidates, 147 satisfy this joint criterion and form the final benchmark.

Relation to full-caption given-action evaluation. The original-input evaluation, conducted before counterfactual construction and final selection, yields 93.53% condition-recovery accuracy for GPT-5.5 on the initial candidate pool. This score summarizes recovery from the original occurrence captions, whereas the final count of 147 reflects successful recovery on both original and counterfactual inputs. The full-caption given-action diagnostic evaluates GPT-5.5 on the 147 retained routines, supplying each target action with the complete Days 1–7 persona history, and yields 57.37% accuracy averaged over three evaluations by GPT-5.6-Luna (Table 4). Both evaluations use identical instructions and output formats, but differ in evidence scope and evaluated routine set. Screening directly supplies the relevant occurrence captions; the full-caption diagnostic additionally requires locating the relevant evidence within the complete history.

A.3 Local Injection Detectability

If synthetic captions are readily recognizable, a model could use injection artifacts to narrow its search for routine occurrences. We therefore test GPT-5.5 on windows of 12 consecutive captions from all personas: half with injections and the rest without. The model is asked to identify an injected caption or report none. We score binary presence, treating any caption selection as predicting presence and a response of none as predicting absence. This measures window-level discrimination rather than correct localization of an injected caption.

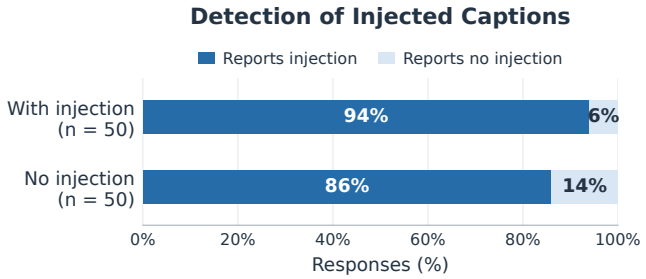


Figure 4: **Injection detectability.** Injection-presence judgments with and without injected captions.

The model reports injections in 94% of windows with injections and 86% of those without (Figure 4), yielding 54% presence accuracy compared with 50% from always predicting presence. Overall, it predicts presence in 90% of windows, showing a strong affirmative response bias. These results indicate that GPT-5.5 fails to reliably distinguish windows with and without injected captions in this local test.

A.4 Dataset Statistics and Ground-Truth Routines

Dataset size and evaluation instances. EgoRoutine-Bench comprises six seven-day caption histories (42 person-days; 195.6 hours; 357,705 captions). Joint discovery is evaluated on these six persona-level histories against 147 ground-truth routines. The given-action diagnostic uses the same 147 routines, with each target action supplied separately. Next-action prediction comprises one

five-choice question per ground-truth routine (147 total), each anchored to a target occurrence on Days 8–9.

Historical occurrence support. The ground truth contains 764 annotated historical occurrences. Each routine has four to fourteen supporting occurrences (mean 5.20; median 5): nine routines have four, 122 have five, eight have six, six have seven, and two have fourteen. Under the caption-replacement construction, these occurrences account for 0.214% of the historical captions. These totals count injected reference occurrences, not the surrounding context captions or all background occurrences considered during reference validation.

Distribution across types and personas. Place (PL), clock time (CT), prior action (PA), and frequency (FR) contain 30, 29, 59, and 29 routines, respectively, supported by 150, 145, 295, and 174 occurrences. Every PL, CT, and PA routine has five occurrences; FR routines have four to fourteen (mean 6.00). For personas A1–A6, the numbers of routines (occurrences) are 25 (136), 22 (112), 25 (127), 25 (136), 25 (126), and 25 (127), respectively. The corresponding mean numbers of occurrences per routine are 5.44, 5.09, 5.08, 5.44, 5.04, and 5.08.

Temporal distribution of support. Across routines, annotated support spans one to seven distinct days (mean 4.54; median 5). A total of 134 routines have occurrences on at least two days; the other 13 are frequency routines whose annotated occurrences fall within a single day. The numbers of supporting occurrences on Days 1–7 are 112, 146, 100, 96, 130, 106, and 74, respectively. These statistics describe the placement of injected occurrences. They do not by themselves establish continuous recording or complete coverage of a day, and do not include background occurrences considered during reference validation.

Complete ground-truth inventory. Table 7 lists all 147 ground-truth routines by recurrence-condition type, together with their occurrence and distinct-day counts. The table preserves the original action and condition wording and the source routine IDs. Appendix A.1 specifies the frequency injection schedules and the tolerance for successive within-day hourly intervals. The occurrence and day counts in this table describe injections, including fourteen occurrences across seven days for each of the two "twice a day" routines. In these IDs, AC, TD, and PF correspond to the paper's PA, CT, and FR labels, respectively; PL is unchanged.

Table 7: **All ground-truth routines in EgoRoutine-Bench, grouped by recurrence-condition type.** Each row preserves the annotated action and condition; occurrence placement and temporal tolerance are detailed in Appendix A.1. n_{occ} counts injected historical occurrences; n_{day} counts distinct days containing those injections. Background occurrences considered during reference validation are excluded from these counts. Routine-ID prefixes J/A/T/L/K/S denote personas A1/A2/A3/A4/A5/A6, respectively.

ID	Action	Recurrence condition	n_{occ}	n_{day}
Place (PL) — 30 routines; 150 occurrences; mean 5.00				
J-PL01	sends a voice message to his mother	in the car	5	5
J-PL02	logs his caffeine intake	at his desk	5	5
J-PL03	does a set of push-ups	in the kitchen	5	5
J-PL04	deletes old screenshots	in his bedroom	5	5
J-PL05	checks sports scores	at the dining table	5	5
A-PL01	records a small expense	in the kitchen	5	5
A-PL02	backs up her photos	in the bathroom	5	5
A-PL03	lays out next-day items	in her bedroom	5	4
A-PL04	checks the weather	at a mirror	5	5
A-PL05	reviews her to-do list	on the sofa	5	5
T-PL01	messages her neighbor	at the supermarket	5	5
T-PL02	knits a few rows	in the living room	5	5
T-PL03	does a breathing exercise	in the car	5	5

Continued on the next page.

Table 7 (continued): Ground-truth routines by type.

ID	Action	Recurrence condition	n_{occ}	n_{day}
T-PL04	calls her daughter	in the kitchen	5	5
T-PL05	practices calligraphy	at the dining table	5	3
L-PL01	checks the school app	in the car	5	4
L-PL02	orders household supplies	in the bathroom	5	5
L-PL03	answers marketplace messages	on the sofa	5	5
L-PL04	reads her horoscope	in her bedroom	5	3
L-PL05	browses a neighborhood forum	in the kitchen	5	5
K-PL01	ticks her habit tracker	at her desk	5	5
K-PL02	browses travel deals	at the dining table	5	5
K-PL03	sketches in her sketchbook	on the sofa	5	5
K-PL04	logs her reading progress	in her bedroom	5	4
K-PL05	checks the pollen count	in the kitchen	5	5
S-PL01	listens to an audiobook	at the stove	5	4
S-PL02	checks his workout plan	in the living room	5	5
S-PL03	checks a fishing forecast	in the car	5	4
S-PL04	works on his stamp album	at the dining table	5	5
S-PL05	messages his hiking club	in the elevator	5	3
Clock time (CT) — 29 routines; 145 occurrences; mean 5.00				
J-TD01	clears his email inbox	late morning (~10-11)	5	5
J-TD02	does a language-app lesson	early afternoon (~13)	5	5
J-TD03	logs his mood in a tracker	late afternoon (~16)	5	5
J-TD04	checks the campus shuttle timetable	evening (~18)	5	5
J-TD05	reads a webtoon episode	night (~21-22)	5	5
A-TD01	checks her budgeting app	late morning (~11)	5	5
A-TD02	adds an item to her grocery list	around noon (~12)	5	5
A-TD03	checks her step count	mid-afternoon (~14-15)	5	5
A-TD04	turns on Do Not Disturb	evening (~19-20)	5	5
T-TD01	browses a deals app	late morning (~10-11)	5	5
T-TD02	checks a plant-care app	around noon (~12-13)	5	5
T-TD03	checks a prescription refill	mid-afternoon (~15-16)	5	5
T-TD04	checks the doorbell camera	early evening (~18-19)	5	5
T-TD05	refills the kettle for the morning	night (~21-22)	5	5
L-TD01	does a vocabulary quiz	late morning (~10-11)	5	5
L-TD02	refills the humidifier	around noon (~12-13)	5	5
L-TD03	checks library holds	mid-afternoon (~15-16)	5	5
L-TD04	browses outfit ideas for tomorrow	early evening (~18-19)	5	5
L-TD05	preps the coffee machine for the morning	night (~21-22)	5	5
K-TD01	browses meetup events	late morning (~10-11)	5	5
K-TD02	browses design inspiration	around noon (~12-13)	5	5
K-TD03	confirms a yoga class booking	mid-afternoon (~14-15)	5	5
K-TD04	puts her earbuds on the charger	evening (~17-18)	5	5
K-TD05	syncs her smartwatch	night (~20-21)	5	5
S-TD01	polishes a pair of shoes	late morning (~10-11)	5	5
S-TD02	books a golf tee time	early afternoon (~13-14)	5	5
S-TD03	organizes his tackle box	late afternoon (~16-17)	5	5
S-TD04	practices his harmonica	evening (~19-20)	5	5
S-TD05	preps his thermos for the morning	night (~21-22)	5	5

Continued on the next page.

Table 7 (continued): Ground-truth routines by type.

ID	Action	Recurrence condition	n_{occ}	n_{day}
Prior action (PA) — 59 routines; 295 occurrences; mean 5.00				
J-AC01	starts the robot vacuum	when picking up his backpack (heading out)	5	5
J-AC02	checks stock prices	when putting on his glasses (settling in to focus)	5	5
J-AC03	sets the thermostat for the night	when plugging in his phone charger (bedtime)	5	5
J-AC04	solves a chess puzzle	when holding his mug (taking a break)	5	5
J-AC05	checks his cloud-storage space	when opening his laptop (starting work)	5	5
J-AC06	messages his study group	after washing the dishes	5	5
J-AC07	checks the mailbox	after taking out the trash	5	5
J-AC08	waters his potted plant	after wiping the table	5	5
J-AC09	adds it to his course notes app	after taking a photo	5	5
J-AC10	browses dinner recipes	after finishing a meal	5	5
A-AC01	sets tomorrow's alarm	when plugging in her charger (bedtime)	5	5
A-AC02	replies in the group chat	when holding her cup/mug (settling in)	5	5
A-AC03	checks her calendar	when putting on her glasses (starting to focus)	5	5
A-AC04	starts a focus timer	when putting on her headphones (focus session)	5	4
A-AC05	checks the shared schedule	when picking up her power bank (heading out)	5	5
A-AC06	writes a journal note	after she finishes eating	5	5
A-AC07	messages a family member	after she takes a photo	5	5
A-AC08	applies lip balm	after opening the refrigerator	5	5
A-AC09	sends a photo to the group	after someone hands her an item	5	5
T-AC01	plays a podcast	when opening her notebook (starting to study)	5	5
T-AC02	bundles old newspapers for recycling	when holding scissors (unpacking/prep)	5	5
T-AC03	reviews her insurance renewal notice	when putting on her glasses (paperwork)	5	5
T-AC04	does a crossword	when holding her cup (taking a break)	5	5
T-AC05	tops up her transit card	when picking up her bag (heading out)	5	5
T-AC06	checks the community-center schedule	when picking up her tablet (leisure)	5	5
T-AC07	opens the window blinds	after pouring tea	5	5
T-AC08	writes tomorrow's menu on the memo pad	after wiping the counter	5	5
T-AC09	texts her husband	after putting on her shoes	5	5
T-AC10	does a set of eye exercises	after finishing a meal	5	3
L-AC01	puts on a cartoon for the kids	when putting a pot on the stove (starting to cook)	5	5
L-AC02	flips through a magazine	when picking up chopsticks (eating alone)	5	5
L-AC03	continues an online course	when sitting down at her computer	5	5
L-AC04	checks her rewards points	when picking up her bag (heading out)	5	5
L-AC05	files the warranty card	when holding scissors (unboxing)	5	5
L-AC06	reads the building notice board	after taking out the trash	5	5
L-AC07	turns on the TV news	after opening the refrigerator	5	5
L-AC08	sprays the air freshener	after wiping the table	5	5
L-AC09	mends clothes (sews a button)	after starting the washing machine	5	3
L-AC10	double-checks the burner knobs	after putting on her shoes	5	5
K-AC01	labels a storage box	when holding scissors (craft/unboxing)	5	5
K-AC02	browses an art-supply shop page	when holding her cup (taking a break)	5	5

Continued on the next page.

Table 7 (continued): Ground-truth routines by type.

ID	Action	Recurrence condition	n_{occ}	n_{day}
K-AC03	drafts a blog post	when picking up her tablet (creative work)	5	5
K-AC04	updates her portfolio site	when sitting down at her computer	5	5
K-AC05	reviews flashcards	when picking up the kettle (making tea)	5	4
K-AC06	turns on the kitchen radio	after opening the refrigerator	5	5
K-AC07	feeds her sourdough starter	after washing the dishes	5	5
K-AC08	reviews her app subscriptions	when picking up the remote (starting to watch TV)	5	5
K-AC09	reads a saved article	after pouring a drink	5	5
K-AC10	posts it to her photo club	after taking a photo	5	5
S-AC01	checks the fine-dust index	when picking up the kettle (making tea/coffee)	5	5
S-AC02	checks flight prices	when putting on his glasses (looking something up)	5	5
S-AC03	runs a virus scan	when sitting at his computer (start of a session)	5	5
S-AC04	browses property listings	when picking up his tablet (leisure)	5	5
S-AC05	starts his foot bath	when picking up the remote (starting to watch TV)	5	3
S-AC06	sharpens a kitchen knife	after wiping the counter	5	5
S-AC07	refills the bird feeder	after taking out the trash	5	5
S-AC08	tears off the daily calendar page	after pouring coffee	5	5
S-AC09	checks its expiry date	after being handed an item	5	5
S-AC10	gargles with salt water	after finishing a meal	5	5
Frequency (FR) — 29 routines; 174 occurrences; mean 6.00				
J-PF01	takes an omega-3 capsule	once per day	7	7
J-PF02	applies pain-relief gel to his shoulder	every other day	4	4
J-PF03	takes a throat lozenge	every 2 hours	6	1
J-PF04	uses ear drops	every 3 hours	5	1
J-PF05	applies acne ointment	twice a day	14	7
A-PF01	takes a vitamin	once per day	7	7
A-PF02	uses a nasal spray	every other day	4	4
A-PF03	puts in eye drops	every 2 hours (while active)	6	1
A-PF04	uses hand sanitizer	every 3 hours (while active)	5	1
T-PF01	takes a probiotic	once per day	7	7
T-PF02	applies a heat patch	every other day	4	4
T-PF03	applies hand lotion	every 2 hours	6	1
T-PF04	uses an inhaler	every 3 hours	4	1
T-PF05	checks her blood sugar	every 4 hours	6	2
L-PF01	takes a fiber gummy	once per day	7	7
L-PF02	applies hair oil	every other day	4	4
L-PF03	applies hand cream	every 2 hours	6	1
L-PF04	applies sunscreen	every 3 hours	5	1
L-PF05	takes a collagen sachet	twice a day	14	7
K-PF01	uses a scalp massager	once per day	7	7
K-PF02	uses a sheet mask	every other day	4	4
K-PF03	checks her heart rate on her fitness band	every 2 hours	6	1
K-PF04	applies cuticle oil	every 3 hours	5	1
K-PF05	uses a lavender roll-on	every 4 hours	4	1
S-PF01	has a red-ginseng stick	once per day	7	7
S-PF02	uses teeth-whitening strips	every other day	4	4

Continued on the next page.

Table 7 (continued): Ground-truth routines by type.

ID	Action	Recurrence condition	n_{occ}	n_{day}
S-PF03	chews nicotine gum	every 2 hours	6	1
S-PF04	applies menthol balm to his neck	every 3 hours	4	1
S-PF05	checks his blood oxygen	every 4 hours	6	2

A.5 Background Routines

The original seven-day EgoLife histories also contain natural routines beyond the injected references. A model that discovers one of these routines should not be penalized simply because it is absent from the reference set. However, these background patterns are not all suitable as additional ground truth. Some have exceptions in the recorded history, and some can be proposed from general knowledge alone. For example, people typically wash their hands at a sink, so a model can suggest this action–place pattern without comparing repeated occurrences in the user's history. Reporting this association therefore does not establish that the model inferred it from the user's behavior.

We account for these background patterns through a separate list of plausible routines, compiled before evaluation from each persona's original history at multiple levels of granularity. For each persona, we exclude every background entry whose action is semantically equivalent to a ground-truth action, regardless of its condition. The resulting neutral inventory is therefore disjoint from the ground truth at the action level. We use GPT-5.6-Luna to semantically match predicted routines against this inventory. Neutral matches receive no credit and are excluded from the precision denominator; they cannot overlap with ground-truth matches or exempt duplicate ground-truth predictions. The resulting scores measure recovery of the injected routines, not discovery of every natural routine. Appendix C.2 specifies the matching and aggregation rules.

B Mathematical Specification of RMI

RMI constructs routine memory through candidate action discovery and pattern induction; a separate predictor reuses this memory for next-action prediction. After candidate action discovery, pattern induction comprises context extraction (context tagging, target-action filtering, and sorting), evidence consolidation (canonicalization and overlap merging), and condition verification. The following subsections specify these stages in order.

B.1 Candidate Action Discovery Settings

Let V index the wearer-action observations extracted in Section 4.1, with source identifier s_i and Qwen3-Embedding-4B embedding $\mathbf{e}_i$ for observation i. Write $\sigma(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}/(\|\mathbf{u}\|_2\|\mathbf{v}\|_2)$ for cosine similarity. Let L_i be the list of up to $K_{\text{obs}} = 128$ retrieved candidates in decreasing order of similarity to i. We exclude self-matches and observations from the same source, require $\tau_{\text{obs}} = 0.94$, and retain at most $k_{\text{obs}} = 4$ neighbors:

$$N_{k_{\text{obs}}}(i) = \text{First}_{k_{\text{obs}}}\big([j \in L_i : j \neq i,\ s_j \neq s_i,\ \sigma(\mathbf{e}_i, \mathbf{e}_j) \geq \tau_{\text{obs}}]\big). \tag{4}$$

Here bracketed filtering preserves list order, and First_k keeps at most the first k entries. The undirected graph $G = (V, E)$ connects observations only when each retains the other:

$$\{i, j\} \in E \iff j \in N_{k_{\text{obs}}}(i) \wedge i \in N_{k_{\text{obs}}}(j). \tag{5}$$

The graph's connected components form the initial observation groups. After temporally overlapping captions are deduplicated within each connected component, the minimum retained support is $m_{\text{occ}} = 2$ occurrence records. This threshold screens candidate actions; it does not establish a recurrence condition. The retained components subsequently undergo recurrence screening and wearer-action projection.

Components with identical canonical action labels are grouped before semantic merge proposals are generated. With group centroids held fixed, each group A examines the $K_{\text{merge}} = 16$ nearest other groups and proposes up to $k_{\text{merge}} = 8$ satisfying cosine similarity $\tau_{\text{merge}} = 0.94$ or higher. A

pair already proposed by another group still counts toward the current group's proposal cap. If S_A denotes the proposed neighbors of group A, the comparison set is

$$\mathcal{U} = \bigcup_A \big\{\{A, B\} : B \in S_A\big\}, \qquad |S_A| \le k_{\text{merge}}. \tag{6}$$

The unordered pair $\{A, B\}$ appears once in $\mathcal{U}$, even if both groups propose it. An LLM applies the `SAME_ACTION` judgment to each comparison pair, and only groups judged to describe the same action are merged. Consequently, a group may participate in more than k_{merge} comparison pairs when other groups propose it; mutual neighbor selection is required for the initial observation graph, not for these merge proposals.

B.2 Context Extraction

Action-conditioned retrieval. For each wearer, RMI indexes the N permitted atomic captions in $\mathcal{M}_{\le T}$ from Days 1–7, preserving their source metadata:

$$\mathcal{B}_{\le T} = \{m_i = (\mathrm{id}_i, d_i, t_i^s, t_i^e, x_i)\}_{i=1}^{N}. \tag{7}$$

Here x_i is the caption, id_i its source identifier, d_i its source day, and $[t_i^s, t_i^e]$ its source interval. N counts caption entries, not distinct action occurrences. For each discovered action a, an LLM generates action-only aliases G_a; these queries do not specify a hypothesized place, time, or contextual trigger. The original action and each alias are encoded with the same dense encoder to rank this index.

Alias fusion and retrieval budget. Let $r_q(i)$ be the zero-based rank of caption m_i for query q. For nonempty G_a, the pooled alias score uses each caption's best rank directly, without treating every alias as a separate fusion branch:

$$\begin{aligned} r_{G_a}(i) &= \min_{g \in G_a} r_g(i), \\ R_a(i) &= \frac{1}{\kappa + r_a(i)} + \frac{1}{\kappa + r_{G_a}(i)}, \qquad \kappa = 60. \end{aligned} \tag{8}$$

When G_a is empty, the second term is omitted. Let $\mathrm{Top}_k(\mathcal{B}; R)$ return at most k captions from $\mathcal{B}$ in descending order of score R. The resulting ordered candidate list is

$$\mathcal{Q}_a^{\text{pre}} = \mathrm{Top}_{30}(\mathcal{B}_{\le T}; R_a). \tag{9}$$

Adding aliases can change the best ranks, but does not add further summands to R_a.

Caption-level target-action filtering. Dense similarity can retrieve related activity without establishing that the wearer performs the target action. A caption-level verifier independently assigns target-action presence $v_a^{\text{cap}}(m) \in \{0, 1\}$ to each retrieved caption, returning only that judgment. In the fused retrieval order, we retain at most 20 positive hits:

$$\mathcal{H}_a = \mathrm{First}_{20}\left([m \in \mathcal{Q}_a^{\text{pre}} : v_a^{\text{cap}}(m) = 1]\right). \tag{10}$$

Lists contain fewer records when fewer are available or pass verification.

Context tagging. Write the retained list as $\mathcal{H}_a = (m_{a,j})_{j=1}^{|\mathcal{H}_a|}$. For each hit $m_{a,j}$, $W_{a,j} = W(m_{a,j})$ concatenates the hit with up to five preceding and five following captions in source order, restricted to the same day and permitted history. The window is truncated at day boundaries; no caption after T is admitted. Atomic captions determine retrieval ranks, while these restored windows supply context for extraction. All windows for action a enter one LLM context-extraction call, which returns a separate structured record for each window $W_{a,j}$:

$$z_{a,j} = (p_{a,j}, \ell_{a,j}, \mathbf{b}_{a,j}, h_{a,j}). \tag{11}$$

The fields are action presence $p_{a,j} \in \{0, 1\}$ (`action_present`), place $\ell_{a,j}$, an ordered action sequence $\mathbf{b}_{a,j}$ with the target step marked, and local state $h_{a,j}$. Window and anchor-caption identities retain the link to source day and timestamps.

Context-level target-action filtering and sorting. The caption-level filter checks short retrieval hits; the context extractor rechecks action presence in the restored windows while supplying the tags used for condition comparison. We retain records with $p_{a,j} = 1$ and sort them by source day and start time:

$$\mathcal{P}_a = \mathrm{Sort}_{\mathrm{time}}\left([(j, W_{a,j}, z_{a,j}) : 1 \le j \le |\mathcal{H}_a|,\ p_{a,j} = 1]\right). \tag{12}$$

Thus, $\mathcal{P}_a$ is an ordered list of action-positive records, with each index j preserving its window's source identity. Only these records proceed to evidence consolidation. Preceding and following actions remain distinguishable through the marked target position: a following action is not evidence for a prior-action condition. Following historical captions can describe an occurrence's context, but are not admissible pre-target evidence when predicting that same occurrence. These are model judgments over captions, not independent verification against video.

B.3 Evidence Consolidation

Canonicalization. An action-specific map ν_a canonicalizes place labels and contextual-action labels separately, choosing each representative from the corresponding observed labels in $\mathcal{P}_a$. Equivalent expressions share a label, while related but distinct places or actions retain different labels. Canonicalization preserves source identity, record order, temporal order within action sequences, the marked target step, and local-state evidence; it does not itself merge events. Write $\nu_a(z)$ for the record obtained by canonicalizing its place and contextual-action fields.

Overlapping-window merging. The canonicalized records retain the chronological order established during context extraction. We merge connected components of overlapping windows on the same day:

$$\mathcal{E}_a = (e_{a,j})_{j=1}^{n_a} = \mathrm{Dedup}_{\mathrm{overlap}}\left([(j, W, \nu_a(z)) : (j, W, z) \in \mathcal{P}_a]\right). \tag{13}$$

$\mathrm{Dedup}_{\mathrm{overlap}}$ forms a graph whose nodes are windows and whose edges connect overlapping windows on the same day, then merges each connected component into one record. The resulting sequence $\mathcal{E}_a$ is chronological, with $n_a = |\mathcal{E}_a| \le |\mathcal{P}_a|$. Thus, windows linked through intermediate overlaps belong to the same component even when they do not overlap directly. This is an operational association heuristic: distinct nearby events can have overlapping context windows, and repeated views need not always overlap. Consequently, the merged records are not certified distinct events.

B.4 Condition Verification

Post-consolidation support requirement. The condition verifier is invoked only if $n_a = |\mathcal{E}_a| \ge 2$: at least two records must remain *after* overlapping windows have been merged. The number of positive records before merging, $|\mathcal{P}_a|$, does not determine eligibility. If overlap merging leaves fewer than two records, the procedure returns `None` without invoking the condition verifier, even when $|\mathcal{P}_a| \ge 2$ before merging. This is a minimum evidence requirement, not a guarantee that the retained records represent distinct events or support a recurrence condition.

Temporal summaries. For eligible actions, temporal gaps are computed from the consolidated records before semantic condition verification. For each retained record $e_{a,j}$, let $\tau_{a,j}$ denote the scalar source timestamp used for gap computation on a common timeline. The condition verifier receives consecutive gaps and compact cross-occurrence summaries:

$$\begin{aligned} \delta_{a,j} &= \tau_{a,j} - \tau_{a,j-1}, && 2 \le j \le n_a, \\ \boldsymbol{\delta}_a &= (\delta_{a,j})_{j=2}^{n_a}. \end{aligned} \tag{14}$$

Condition inference and verification. Let $\mathcal{C}_{\mathrm{cond}}$ denote the space of concrete conditions under the prompt's four types: clock time, periodicity, place, and adjacent action. The LLM condition verifier infers a candidate condition from the supplied records and assesses its support under these definitions, returning a concrete condition or `None`. Write $\perp$ for `None` and $\mathrm{Stats}(\mathcal{E}_a)$ for the precomputed gaps and cross-record summaries. The action-specific output $\hat{\rho}_a \in \mathcal{C}_{\mathrm{cond}} \cup \{\perp\}$ is

$$\hat{\rho}_a = \begin{cases} \mathrm{VerifyCondition}\big(a, \mathcal{E}_a, \mathrm{Stats}(\mathcal{E}_a)\big), & n_a \ge 2, \\ \perp, & n_a < 2. \end{cases} \tag{15}$$

Here, $\mathrm{VerifyCondition}$ denotes the condition verifier's inference-and-verification call; it compares only the supplied evidence and does not retrieve additional history.

Routine memory retention. The retained routine memory contains at most one non-null condition per candidate action:

$$\widehat{\mathcal{R}}_{\leq T} = \big((a, \hat{\rho}_a)\big)_{a \in \mathcal{A}: \hat{\rho}_a \neq \perp}. \tag{16}$$

Each retained pair is expressed as a statement scoped to the observed history. The local-state field supplies contextual evidence rather than a separate condition family. Abstention omits a memory entry without proving the absence of a routine.

Relation to the reference criteria. A match to a prior-action reference requires the contextual action to precede the target, regardless of the method's adjacent-action label. Likewise, gaps between retrieved occurrences support an "every N hours" frequency condition only if the occurrences are verified to be successive recorded events within the same daily observation window and satisfy Equation 3; count-based conditions require counts of all recorded occurrences within their evaluation periods. Capped retrieval and overlap consolidation do not guarantee complete or distinct-event coverage, so a condition selected from $\mathcal{E}_a$ need not satisfy the full-history reference criteria in Appendix A.1.

C Experimental Protocols

C.1 Baseline Variants and Resource Settings

Discovery variants. *Full captions* supplies the entire historical caption stream directly to the backbone. The memory-agent baselines are EgoRAG (Yang et al., 2025), Ego-R1 (Tian et al., 2026), WorldMM (Yeo et al., 2026), EgoMemo (Sitong et al., 2026), and M3-Agent (Long et al., 2026). Both baseline families use two discovery variants: *Direct* requests an action–condition collection in one step, whereas *Action-first* first requests candidate actions and then their recurrence conditions. Neither variant receives target actions or the number of routines to discover. This comparison tests whether explicit action enumeration improves discovery.

Prediction variants. *Full captions* uses the complete Days 1–7 history. For each memory agent, *Ret.* uses evidence from its original retrieval pipeline, while *Disc.* uses routine memory converted from its *Action-first* discovery outputs. All discovered routine collections, including RMI's, undergo the same conversion before being supplied to the predictor. The *None* and *Oracle* controls supply no historical memory and ground-truth routines, respectively. All configurations receive the current context and five-choice question. Appendix C.3 formalizes these inputs and their shared historical cutoff.

Backbones and resource settings. Motivated by on-device assistance, we evaluate RMI and the memory-agent baselines with Qwen3.5-2B and Qwen3.5-4B. Full-caption discovery additionally includes Claude Sonnet-5, Gemini-3.5-Flash, and GPT-5.5. For both Qwen backbones, all methods use the same default context limit of 65K tokens (`max_model_length`), with YaRN scaling (factor 4) enabled when generation requires a longer context. Memory-agent baselines are allowed at most five retrieval calls per query, whereas RMI performs one.

C.2 Semantic Matching and Metric Aggregation

Matching unit and eligibility. We use GPT-5.6-Luna to compare each predicted action–condition pair with the ground-truth routines of the same persona. The judge applies the action and condition criteria below before selecting credited predictions. *Action* matching requires the same action; *Joint* matching requires both that action and the condition of the same reference routine. A high matching score cannot override the eligibility criteria for the metric being computed; condition agreement is required for Joint matching only.

Action identity and object specificity. Different wording is allowed when it describes the same behavior with the same semantic scope. If the reference action specifies an object, the predicted action must preserve that object semantically, together with the action performed on it. An omitted object, a different object, or a broader object category does not qualify merely because the verb is similar. For

example, "removes old screen captures" can match "deletes old screenshots," whereas "deletes old files" is too broad. These are semantic-equivalence decisions, not requirements for identical strings.

Condition matching. For PL and CT, the judge checks all verified distinct occurrences of the target action in the final Days 1–7 history, including injected and background occurrences. The criteria for each recurrence family are:

- **Place (PL).** Equivalent expressions such as "in the kitchen" and "inside the kitchen" match. A more specific location is accepted only when evidence supports it for every verified occurrence. Thus, "at the kitchen counter" matches "in the kitchen" only if evidence places all verified occurrences at that counter; an occurrence elsewhere or insufficient evidence of the added constraint precludes a match. "In the bathroom" and the broader "at home" do not match "in the kitchen."
- **Prior action (PA).** The preceding action must match semantically, including its object when specified, and its precedence over the target action must be preserved. For example, "after wiping the table's surface" can match "after wiping the table"; "after wiping the counter" and "after cleaning" do not.
- **Clock time (CT).** Labels such as "early afternoon (~13)" denote approximate time-of-day windows. A predicted window matches if it is no broader than the ground-truth window and covers the clock times of all verified occurrences of the action. A wider window or one excluding any verified occurrence is rejected. No endpoint tolerance is used.
- **Frequency (FR).** The recurrence relation and associated unit or period must match after normalizing equivalent units, such as every two hours and every 120 minutes. For "every N hours," the predicted interval must exactly equal the reference interval. Day-based intervals and counts must match exactly, including "every other day," "once per day," and "twice a day." "Twice a day" is a count per day and is not equivalent to "every 12 hours," which imposes a spacing constraint.

The ± 5-minute tolerance in Appendix A.1 applies only to validating recorded occurrence gaps against annotated hourly intervals. The same condition rubric applies to given-action condition induction, where the reference action is already supplied.

One-to-one matching and duplicate predictions. Within each persona, the ground-truth actions are semantically distinct under the action-and-object criterion, with one reference condition per action. Each prediction can therefore be assigned to at most one ground-truth routine; it cannot earn credit for several references through a broad action description. For each metric separately, we first identify the predictions satisfying that metric's eligibility criteria. If several eligible predictions match the same reference, only the one with the highest judge-assigned matching score is retained as a true positive (TP). All other eligible matches to that reference are duplicate false positives (FPs), even if they use different wording. Duplicates remain in the precision denominator and provide no additional recall credit; at most one TP is counted per reference, including when matching scores tie. A prediction with the correct action but an incorrect condition is eligible for Action matching but not Joint matching. It is a Joint FP, and the reference is also a Joint false negative (FN) if no other prediction correctly matches the complete routine.

Neutral predictions and mutual exclusivity. The persona-specific background inventory is fixed before evaluation and excludes every entry whose action is semantically equivalent to a ground-truth action, regardless of its condition (Appendix A.5). A neutral decision requires both the predicted action and condition to match an entry in this inventory under the same semantic and specificity criteria. Ground-truth and neutral matches are mutually exclusive at the action level. In particular, a prediction with a ground-truth action and an incorrect condition cannot be exempted as neutral, and a duplicate ground-truth prediction cannot be reclassified as neutral. Neutral matches receive no TP credit, are excluded from the precision denominator, and leave the ground-truth recall denominator unchanged. Neutral filtering applies only to discovery scoring; it does not remove entries from the routine memory used for next-action prediction.

Aggregation and prediction accuracy. For each metric, every prediction is counted exactly once as a retained TP, a duplicate or unmatched FP, or a neutral exclusion. Every ground-truth routine

without a retained match is an FN. We sum TP, FP, and FN counts across the six persona histories to compute micro-averaged precision, recall, and F1, with joint F1 as the primary metric; the discovery tables report averages over three runs. Next-action accuracy is the percentage of correctly selected answers across the 147 five-choice questions on Days 8–9, covering all ground-truth routines.

C.3 Next-Action Prediction Inputs

Target occurrences and context prefixes. Each question is constructed around a designated occurrence of a ground-truth routine in Days 8–9. For PL, CT, PA, and day-based FR routines ("once per day," "twice a day," and "every other day"), the current context c_j is the prefix of the Days 8–9 caption stream ending immediately before the target action's first occurrence. Only hourly-interval FR routines ("every N hours") use the prefix ending immediately before the second occurrence, retaining the first occurrence as a temporal reference. For day-based FR routines, earlier occurrences are already present in the Days 1–7 history. Here, "first" and "second" count occurrences within the Days 8–9 caption stream. The target occurrence and all subsequent captions are excluded from c_j. The evaluation therefore measures action selection at designated routine occurrences, rather than at arbitrarily sampled future timestamps.

Choice construction and answer validation. Each multiple-choice question contains the target ground-truth action and four distractor actions sampled at random from the other actions in the same persona's ground-truth routine list. Correct-answer positions are distributed as evenly as possible across the five option slots over the evaluation set. Human evaluation and a GPT-5.5 judge checked each question and its five choices against the current context and ground-truth routines, confirming that it has exactly one correct answer and that no distractor is also a valid answer.

RMI constructs $\widehat{\mathcal{R}}_{\le T}$ from Days 1–7 and holds it fixed across prediction questions. For the selected target at $t_j > T$, reuse is

$$\hat{a}_j = P(c_j, \mathcal{C}_j, \widehat{\mathcal{R}}_{\le T}), \qquad |\mathcal{C}_j| = 5. \tag{17}$$

This avoids repeating historical induction for every prediction. This temporal separation is shared by all discovered routine-memory inputs; oracle references are a separate input condition.

For method m, let $\mathcal{H}_{m,\le T}$ denote its historical memory and q_j the prediction question text. Write $Q_j = (q_j, \mathcal{C}_j)$ for the question together with its five candidate answers. The retrieval request is formed from the current context c_j and Q_j; the method may expand this request into choice-specific searches or model-generated queries according to its retrieval implementation. If $u_{m,j}$ denotes the resulting query or query set and $E_{m,j}$ the retrieved memory, the *Ret.* input is

$$\begin{aligned} u_{m,j} &= \mathrm{Query}_m(c_j, Q_j), \\ E_{m,j} &= \mathrm{Retrieve}_m(\mathcal{H}_{m,\le T}, u_{m,j}), \\ x^{\mathrm{Ret.}}_{m,j} &= \mathrm{Append}(c_j, Q_j, E_{m,j}). \end{aligned} \tag{18}$$

The answer model receives the current context and original question together with the retrieved memory, and selects one of the five candidates.

For each baseline method m, let $\widehat{\mathcal{R}}_{m,\le T}$ be its routine collection from *Action-first* joint discovery; for RMI, this is the retained collection in Equation 16. Writing g_{mem} for the common routine-output-to-memory conversion gives

$$\begin{aligned} U_m &= g_{\mathrm{mem}}\left(\widehat{\mathcal{R}}_{m,\le T}\right), \\ x^{\mathrm{Disc.}}_{m,j} &= \mathrm{Append}(c_j, Q_j, U_m). \end{aligned} \tag{19}$$

The same g_{mem} is used for every method, including RMI; only its input routine collection is method-specific. The resulting memory U_m is appended to the current context and question before answer generation, with the answer-model backbone listed in Table 3. The Oracle input instead appends ground-truth routines. Prediction memory uses discovery outputs obtained without supplied target actions; reference labels must not filter this memory, and given-action diagnostic outputs are excluded.

The accuracy comparison in Section 5.2 evaluates the predictive utility of these inputs; it does not measure their online input-token or total computational costs. A total-cost comparison would additionally account for constructing and updating the historical representation over the number of predictions using it.

D Additional Experiments

D.1 Full Joint-Discovery Results

Table 8 reports the complete Qwen3.5-4B and Qwen3.5-2B results corresponding to the main-paper comparison.

Table 8: **Full joint action–condition discovery results without supplied target actions.** Values are averaged over three runs. Scores use GPT-5.6-Luna judgments. Action evaluates action matching; Joint additionally requires a matching recurrence condition. Both report precision (P), recall (R), and F1 in percent. Discovered Routines counts predictions excluding semantic matches to the neutral list. Full-caption Qwen runs use YaRN factor 4.

Backbone	Method	Discovery protocol	Discovered Routines	Action			Joint		
				P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑
With full captions									
Qwen3.5-2B	Full captions	Direct	17.00	11.79	1.36	2.44	0.00	0.00	0.00
		Action-first	54.00	15.48	5.67	8.29	0.00	0.00	0.00
Qwen3.5-4B	Full captions	Direct	4.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	8.00	25.26	1.36	2.58	0.00	0.00	0.00
Claude Sonnet-5	Full captions	Direct	50.33	53.19	17.92	26.74	18.72	6.35	9.46
		Action-first	72.67	58.19	28.57	38.30	19.76	9.75	13.05
Gemini-3.5-Flash	Full captions	Direct	30.00	87.96	17.92	29.75	71.46	14.52	24.11
		Action-first	57.00	85.42	32.88	47.44	43.78	16.78	24.24
GPT-5.5	Full captions	Direct	51.33	92.05	31.97	47.35	46.31	15.88	23.59
		Action-first	136.67	83.54	77.55	**80.42**	46.85	43.54	**45.12**
Memory agents									
Qwen3.5-4B	EgoRAG (Yang et al., 2025)	Direct	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	105.33	3.81	2.72	3.17	0.95	0.68	0.79
	Ego-R1 (Tian et al., 2026)	Direct	12.33	16.24	1.36	2.51	8.12	0.68	1.26
		Action-first	38.00	35.09	9.07	14.41	5.27	1.36	2.16
	WorldMM (Yeo et al., 2026)	Direct	11.67	17.25	1.36	2.52	0.00	0.00	0.00
		Action-first	10910.67	0.93	68.93	1.84	0.51	37.64	1.00
	EgoMemo (Sitong et al., 2026)	Direct	330.33	0.30	0.68	0.42	0.30	0.68	0.42
		Action-first	3124.67	0.21	4.53	0.41	0.18	3.85	0.35
	M3-Agent (Long et al., 2026)	Direct	168.00	2.40	2.72	2.54	0.60	0.68	0.64
		Action-first	667.33	1.65	7.48	2.71	0.60	2.72	0.99
	RMI (ours)	Action-first	209.00	29.04	41.05	**33.98**	26.23	37.19	**30.73**
Qwen3.5-2B	EgoRAG (Yang et al., 2025)	Direct	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	310.33	0.65	1.36	0.88	0.65	1.36	0.88
	Ego-R1 (Tian et al., 2026)	Direct	7.00	28.57	1.36	2.60	0.00	0.00	0.00
		Action-first	80.33	32.43	17.69	**22.86**	2.51	1.36	1.76
	WorldMM (Yeo et al., 2026)	Direct	13.00	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	3731.00	2.79	70.30	5.38	0.56	14.29	1.08
	EgoMemo (Sitong et al., 2026)	Direct	90.67	0.00	0.00	0.00	0.00	0.00	0.00
		Action-first	2693.67	0.26	4.76	0.49	0.15	2.72	0.28
	M3-Agent (Long et al., 2026)	Direct	302.67	1.00	2.04	1.34	0.00	0.00	0.00
		Action-first	821.33	0.85	4.76	1.45	0.25	1.36	0.41
	RMI (ours)	Action-first	524.67	13.89	49.21	21.62	7.96	28.34	**12.41**

D.2 JUDGE EXPANSION EXPERIMENT

Table 9 tests sensitivity to evaluator choice using GPT-5.6-Luna, Qwen3.8-27B, Claude Sonnet-5, and Gemini-3.5-Flash. Each judge column averages three evaluations by that judge; the Average column reports the arithmetic mean of these four judge-specific averages. Table 4 summarizes the GPT-5.6-Luna results.

Table 9: **Judge expansion for given-action condition induction.** Each judge column reports accuracy averaged over three evaluations by that judge. The Average column is the arithmetic mean of the four judge-specific averages. Table 4 reports the GPT-5.6-Luna column. Ground-truth actions are supplied, and accuracy is reported as a percentage.

Backbone	Method	Judge model / Acc. (%) ↑				Average (%) ↑
		GPT-5.6 Luna	Qwen3.8 27B	Claude Sonnet-5	Gemini-3.5 Flash	
With full captions						
GPT-5.5	Full captions	57.37	57.82	56.69	60.54	58.11
Sonnet-5	Full captions	31.75	31.29	32.65	30.61	31.58
Gemini-3.5-Flash	Full captions	38.78	39.46	38.78	40.14	39.29
Qwen3.5-4B	Full captions	10.20	17.69	18.37	13.15	14.85
Qwen3.5-2B	Full captions	3.40	4.76	2.04	4.76	3.74
Memory agents						
Qwen3.5-4B	EgoRAG (Yang et al., 2025)	13.15	13.83	12.24	14.29	13.38
	Ego-R1 (Tian et al., 2026)	12.93	13.15	13.83	12.93	13.21
	EgoMemo (Sitong et al., 2026)	33.11	30.61	33.33	32.65	32.43
	WorldMM (Yeo et al., 2026)	55.33	54.20	56.69	55.33	55.39
	M3-Agent (Long et al., 2026)	42.86	42.18	42.18	41.50	42.18
	RMI (ours)	**76.42**	**79.14**	**74.60**	**78.91**	**77.27**
Qwen3.5-2B	EgoRAG (Yang et al., 2025)	12.93	13.15	12.93	13.15	13.04
	Ego-R1 (Tian et al., 2026)	12.24	11.56	12.24	12.93	12.24
	EgoMemo (Sitong et al., 2026)	14.29	12.93	11.56	13.61	13.10
	WorldMM (Yeo et al., 2026)	42.63	41.50	40.82	42.18	41.78
	M3-Agent (Long et al., 2026)	30.61	31.29	29.93	29.93	30.44
	RMI (ours)	**61.90**	**61.90**	**64.17**	**65.99**	**63.49**