MEMORA: Embodied Action Memory from Egocentric Videos for Reasoning and Planning

Zihao Yu[1] **Xiu Yuan**[1] **Chongjie Zhang**[1]
[1]Washington University in St. Louis
{yu.zihao,xiu,chongjie}@wustl.edu

Abstract

Embodied agents accumulate experience over time. We study how accumulated experience can be formed into persistent memory for future reasoning and action. We formulate Embodied Action Memory (EAM) as the capability to form and use memory over embodied experience, together with the persistent memory state produced by that process. We introduce MEMORA, a framework that instantiates EAM through a formation–consolidation–retrieval lifecycle and a multi-store world-memory architecture. MEMORA organizes experience into participant-specific Environment, Entity, Activity, and Inferred Knowledge stores: online editing revises memory as new evidence arrives, while offline consolidation abstracts repeated experience into reusable routines, habits, and preferences. We evaluate MEMORA with MEMORA-Bench, a 45-hour egocentric-video suite that measures both retrospective memory faithfulness and prospective memory-grounded planning. Across four open-weight answer models, MEMORA achieves the strongest aggregate planning performance among the evaluated memory interfaces, with its largest gains on out-of-distribution planning. On these tasks, MEMORA improves Robot-Grounded Plan score by up to 16.6%, suggesting that memory formed and consolidated across experience can support planning for new goals beyond directly observed episodes. A physical-robot demonstration further shows that memory formed solely from human egocentric video can ground high-level robot plans in participant-specific objects and preferences.

1 Introduction

Human intelligence is not only a response to the present; it is shaped by experience accumulated across time. Memory turns such experience into a persistent basis for future reasoning and action (Tulving, 1972; McClelland et al., 1995; Stickgold, 2005). We study embodied agents through the same lens: their later decisions can depend not only on the current observation, but also on memory formed from prior embodied experience.

A long tradition in cognitive and computational models treats memory not as a passive archive, but as a staged process of encoding, consolidation, and retrieval, often across functionally distinct forms of memory (Atkinson and Shiffrin, 1968; Tulving, 1972; McClelland et al., 1995). This lifecycle view has shaped classical cognitive architectures (Anderson, 1983; Laird, 2012), neural agents with differentiable read–write memory (Weston et al., 2015; Graves et al., 2016), and recent LLM-agent systems that explicitly model memory formation and consolidation (Ma et al., 2026; Mody et al., 2026; Ye et al., 2026; Du, 2026). Embodied settings complicate this lifecycle because a continuous perception stream interleaves different kinds of information that persist, change, and become useful in different ways, including spatial relations among places and objects, entities tracked across changing states, temporally ordered actions, and regularities that emerge only across episodes. Forming memory from this stream involves revising earlier content as evidence arrives and consolidating repeated experience while preserving distinctions that later reasoning and action may draw upon. We call this capability **Embodied Action Memory (EAM)**: the ability to form and use memory over embodied experience, together with the persistent memory state produced by that process.

We therefore study embodied memory as a formation–consolidation lifecycle (Figure 1) that maintains changing experience online and abstracts repeated experience offline. Egocentric video provides a longitudinal first-person record of how a participant interacts with familiar environments. Whereas prior work often uses this signal for imitation or cross-embodiment policy learning (Kareer et al., 2024; Punamiya et al., 2026; Zheng et al., 2026a), MEMORA uses it to form participant-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 25551–25600
October 24-29, 2026 ©2026 Association for Computational Linguistics

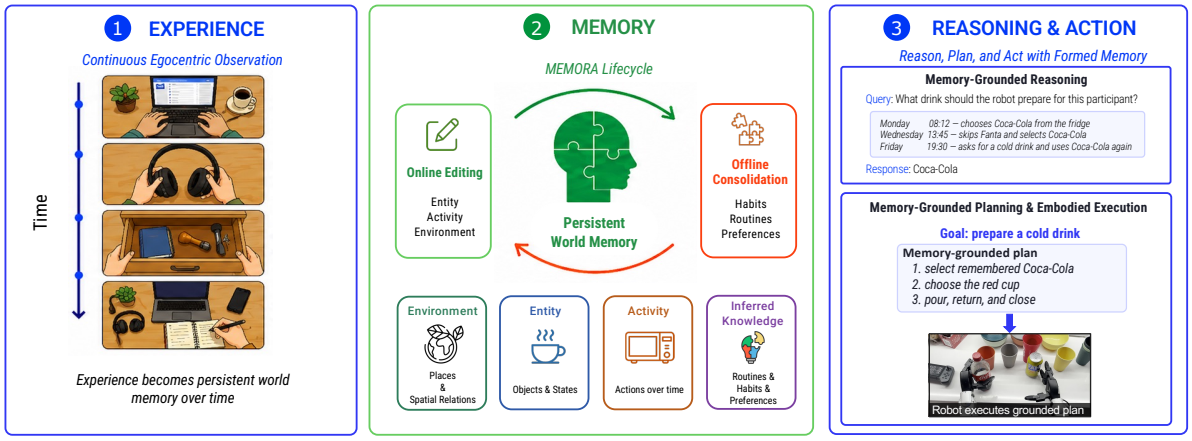


Figure 1: **MEMORA studies embodied memory as a lifecycle from experience to action.** An agent observes continuous egocentric experience, forms typed memory online, and consolidates repeated experience into reusable routines, habits, and preferences. The resulting persistent world memory supports memory-grounded reasoning, planning, and physical robot execution from accumulated experience.

specific persistent memory.

We propose **MEMORA** (Figure 1), an embodied memory framework for forming, consolidating, and using persistent world memory from continuous egocentric experience. To preserve distinctions among these forms of embodied information, MEMORA organizes memory into four typed stores: **Environment Memory** for spatial context, **Entity Memory** for object identity and state, **Activity Memory** for temporally ordered action traces, and **Inferred Knowledge** for regularities consolidated across repeated experience. During online formation, an active Memory Editor applies LLM judgment to entity observations, selecting ADD, UPDATE, DELETE, or NOOP and yielding a median $18\times$ compression while preserving state histories. After offline consolidation, typed retrieval at use time separates reusable procedure from physical grounding, letting MEMORA compose reusable routines with participant-specific entities into grounded language plans, even for goals never directly observed.

To evaluate this view, we introduce **MEMORA-Bench**, a 45-hour egocentric-video evaluation suite built from an EPIC-KITCHENS extension split (Damen et al., 2022). It pairs **MEMORA-Embodied Memory Assessment**, which tests whether constructed memory is faithful to prior experience, with **MEMORA-Planning**, which tests whether memory supports future action. Planning includes **Replay** tasks with previously observed workflows and **Generalize** tasks requiring transfer, composition, or new goals. Across four open-weight backbones, MEMORA improves most on out-of-distribution planning, reaching a $+16.4\%$ relative gain in Robot-Grounded Plan score (RGP) on Gemma-4-31B-it. A physical-robot demonstration further shows that memory formed solely from human egocentric video can ground high-level robot plans in participant-specific objects and preferences. MEMORA therefore complements the robot's execution system: memory helps determine and ground what to do, while an embodiment-specific controller carries out the resulting plan.

Contributions. (1) A formulation of **Embodied Action Memory (EAM)** (§3) as the formation and use of persistent memory over embodied experience, separating how memory is formed from how it later supports reasoning and action. (2) **MEMORA** (§4), an embodied memory framework that instantiates the formation–consolidation–retrieval lifecycle on real egocentric video through four typed stores—**Environment Memory**, **Entity Memory**, **Activity Memory**, and **Inferred Knowledge**—specialized for different forms of embodied information and their update dynamics, together with write-time active editing (median $18\times$ compression), offline consolidation, and typed retrieval at use time. (3) **MEMORA-Bench**, an evaluation suite for EAM over 45 hours of egocentric video, pairing memory faithfulness (**MEMORA-Embodied Memory Assessment**) with memory-grounded action (**MEMORA-Planning**). (4) A **systematic empirical study** of complete memory interfaces across four answer backbones, with within-family contrasts isolating online editing and offline consolidation; the largest planning gains occur out of distribution.

2 Related Work

Memory formation. Agent memory has been modeled as parametric storage in model weights (Zhang et al., 2024) or as explicit stores retrieved as documents, embeddings, summaries, graphs, or edited records (Lewis et al., 2021; Zhong et al., 2023; Packer et al., 2024; Park et al., 2023; Chhikara et al., 2025; Yan et al., 2026). M3-Agent, for example, organizes multimodal video experience as a graph for long-term remembering and reasoning (Long et al., 2025). Recent work introduces explicit encoding and consolidation over text or simulated experience (Ma et al., 2026; Mody et al., 2026; Ye et al., 2026; Du, 2026), reflecting cognitive distinctions between episodic, semantic, and consolidative memory (Tulving, 1972; McClelland et al., 1995; Stickgold, 2005; Nader, 2003). MEMORA brings this direction to continuous egocentric video, where memory is revised online and consolidated into persistent state. MEMORA thus differs not only in how past experience is retrieved, but also in treating memory formation itself as part of the embodied problem.

Embodied memory benchmarks. Long-video QA benchmarks evaluate comprehension of video supplied at query time (Fu et al., 2025; Mangalam et al., 2023; Li et al., 2024), while conversational-memory benchmarks isolate memory in text-only interactions (Maharana et al., 2024; Wu et al., 2025). Egocentric datasets and instruction-following suites provide rich embodied activity, but these resources do not evaluate whether an agent can form memory across sessions and later use it without receiving the relevant video at test time (Damen et al., 2022; Grauman et al., 2022; Shridhar et al., 2020; Padmakumar et al., 2021). MEMORA-Bench targets this experience-to-memory setting: questions and plans are grounded in a participant's accumulated world and require evidence about preferences, routines, object choices, and spatial context across sessions.

Memory-grounded embodied action. Language-conditioned planning uses prompted LLMs, code, or skill libraries to connect perception, reasoning, and action (Zeng et al., 2022; Ahn et al., 2022; Liang et al., 2023; Huang et al., 2022). Other embodied agents maintain spatial memory or learn reactive vision–language–action policies for grounding and control (Yang et al., 2025; Zheng et al., 2026b; Lei et al., 2026; Brohan et al., 2023; Black et al., 2026). Egocentric video also supports imitation, manipulation, and cross-embodiment learning (Kareer et al., 2024; Punamiya et al., 2026; Zheng et al., 2026a); MEMORA instead uses the same longitudinal signal to form persistent participant memory for downstream reasoning and action. It supplies participant-specific context to downstream planners and policies, complementing spatial maps and visuomotor capabilities without replacing their metric grounding or control machinery.

3 Problem Formulation: Embodied Action Memory

Embodied intelligence unfolds over time: observations arrive before their future relevance is known. **Embodied Action Memory (EAM)** studies how temporally extended embodied experience can be formed into a persistent, evolving memory state that can later support reasoning and action. Let o_t be the observation at time t, $\mathcal{M}_t$ the memory state after processing experience through t, and x_τ a later question, goal, or decision context. We model experience formation and later use as

$$\mathcal{M}_t = F(\mathcal{M}_{t-1}, o_t), \qquad d_\tau = G(\mathcal{M}_\tau, x_\tau), \tag{1}$$

Here, F maps incoming embodied experience to an evolving memory state, while G uses the memory available at time τ together with a later decision context to produce an answer, plan, or other action-relevant decision. This formulation separates how memory is formed from how it is later used, without prescribing a single memory representation or decision interface. In this work, we instantiate EAM with longitudinal egocentric experience, typed memory stores, and memory-grounded reasoning and planning.

4 Method

We present MEMORA along the EAM lifecycle: §4.1 describes online formation and offline consolidation, §4.2 exposes the formed memory for retrieval and planning, and §4.3 introduces MEMORA-Bench. Online encoding and active editing instantiate the online formation function F, while offline consolidation abstracts repeated experience into reusable knowledge (Ma et al., 2026; Mody et al., 2026; Ye et al., 2026). To address heterogeneous forms of information and their update dynamics in embodied experience, MEMORA

combines four typed stores with *write-time active editing*.

4.1 Memory Formation: Encoding, Editing, and Consolidation

MEMORA instantiates the EAM formulation in Section 3 with four typed stores: **Environment Memory**, **Entity Memory**, **Activity Memory**, and **Inferred Knowledge** (Table 1).

The design distinguishes four forms of embodied information: recurring spatial context, persistent object identities and changing states, temporally ordered action evidence, and routines or preferences that emerge across repeated experience. A single append-only store flattens these distinctions, while a single graph-structured store can blur them. Motivated by both the multi-store view of memory (Atkinson and Shiffrin, 1968) and the functional distinction between episodic traces and semantic generalizations (Tulving, 1972; McClelland et al., 1995), MEMORA therefore maintains a *multi-store memory state* $\mathcal{M}_t = (\mathcal{M}_t^{\text{env}}, \mathcal{M}_t^{\text{ent}}, \mathcal{M}_t^{\text{act}}, \mathcal{M}^{\text{inf}})$ (Table 1), with stores specialized for these forms of information and their update dynamics; this typing is not an additional supervision signal but determines the update rule each observation receives, concentrating LLM judgment where semantic revision is necessary and using deterministic operations where the rule is stable.

During *online memory formation*, the write path first encodes the current segment with working context, then revises the persistent memory state via EDIT, which instantiates the formation function F from Section 3:

$$\begin{aligned}(z_t, c_t) &= \phi(v_t, c_{t-1}),\\ \mathcal{M}_t &= \textsc{Edit}(\mathcal{M}_{t-1}, z_t).\end{aligned} \tag{2}$$

Here ϕ is the *Segment Encoder*, a multimodal model that converts each 10 s video segment v_t into text-structured observations $z_t = (z_t^{\text{env}}, z_t^{\text{ent}}, z_t^{\text{act}})$, and c_t carries a short working-memory summary across segment boundaries. Inside EDIT, following recent active memory-editing systems (Chhikara et al., 2025; Yan et al., 2026), the *Memory Editor* selects $\delta_j \in \Delta = \{\textsc{Add}, \textsc{Upd}, \textsc{Del}, \textsc{Noop}\}$ per object observation while preserving previous states in `state_history`; **Environment Memory** and **Activity Memory** are updated deterministically (merge by detected place; append by segment), and **Inferred Knowledge** is populated offline by consolidation (Stickgold, 2005) over repeated entity and activity evidence. Later evidence can update histories or delete inconsistent identifications, so editing revises earlier memory rather than only compressing it. JSON-style records, entity-edit diagnostics, and memory-store composition are in Appendix A.

MEMORA thus maintains memory selectively rather than storing every observation as an independent record; Figure 2 quantifies a median $18\times$ reduction relative to the unedited observation stream on the 18-participant EPIC-Kitchens corpus (state histories preserved).

4.2 Memory-Guided Retrieval and Planning

Once memory has been formed, MEMORA exposes it through two uses of the decision function G: retrospective assessment and prospective planning. *Retrieval* is an evaluation interface: given a question q and the completed memory bank $\mathcal{M}^{(p)}$ for its target participant p, the agent is asked to recover participant-specific evidence and select an answer a only when that memory supports it. *Planning* is an application interface: given a goal g, the agent is asked to produce a grounded plan $\pi = ((a_1, e_1, \ell_1), \ldots, (a_n, e_n, \ell_n))$ from the memory state, where each step specifies an action a_k, an entity e_k, and a location ℓ_k drawn from memory.

Both evaluations use an iterative ReAct loop (Yao et al., 2023), with read interfaces matched to each memory representation. Each condition exposes a common search entry point, while Graph-2D and MEMORA additionally provide representation-specific temporal or typed-store helpers; MEMORA's planning interface also exposes consolidated routines and preferences. Cross-family results compare complete memory interfaces; within-family controls are specified in §5.1. At read time, planners bind procedural scaffolds from **Inferred Knowledge**/**Activity Memory** ("how to do it") to participant-specific entities, states, and locations from **Entity Memory**/**Environment Memory** ("which object and where"), especially on Generalize, where planning composes evidence across memories. The released tool surface, participant routing, helper semantics, retrieval settings, prompts, and iteration budgets are specified in Appendix F; memory-bank construction is detailed in Appendix A, and an end-to-end Planning trace appears in Appendix C.

Store	Record unit	Example	Read-time role
Environment Memory	Place records with layout descriptions, named zones, and spatial relations.	`sink_area`: zones {`sink`, `dish_rack`, `stove`, `faucet`}; `dish_rack` to the right of `sink`.	Ground actions and objects in remembered places.
Entity Memory	Object records with visual attributes, location, current state, and `state_history`.	`plate_with_food_residue` (white ceramic, dirty): UPD *idle@countertop* → *being_washed@sink*; prior state retained in `state_history`.	Preserve object identity through changing state.
Activity Memory	Segment-aligned action records (10 s segments) with summary, narrative, and `action_breakdown`.	`10–20 s` "picks up pan and washes it in the sink"; `action_breakdown`: 10–12 s picks up pan, 12–14 s moves to sink, 14–20 s washes pan.	Recover event evidence, temporal order, and procedure traces.
Inferred Knowledge	Routines, habits, preferences, and workflow patterns linked to supporting episodes.	Workflow pattern *"dirty dishes and used containers are washed in the sink, then placed on the countertop"* (supports: `pan_black`, `plate_with_food_residue`; conf. 0.95).	Support cross-event generalization and planning routines.

Table 1: **Typed memory stores in MEMORA.** Examples are drawn from a real kitchen recording (participant P09).

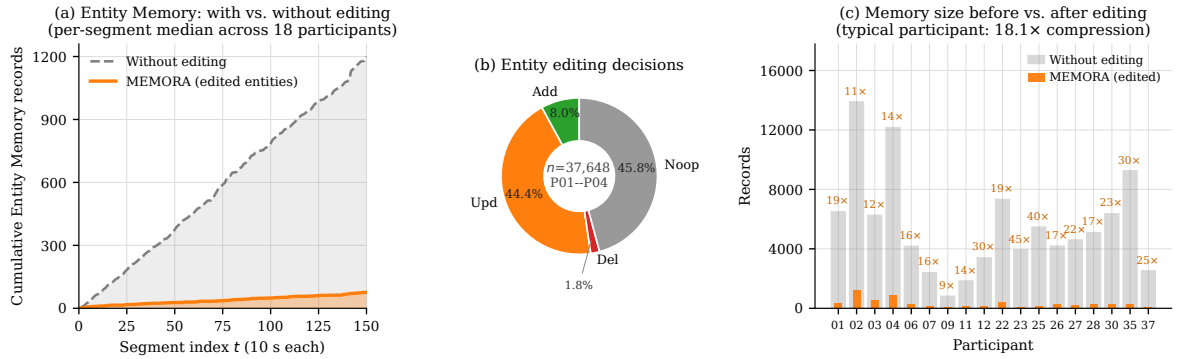


Figure 2: **Active memory formation in Entity Memory (18 participants). (a)** Cumulative records with and without the editor (per-segment median, first 150 segments). **(b)** Editor decision mix (n=37,648, P01–P04): NOOP (45.8%) and UPD (44.4%) dominate ADD (8.0%) with rare DEL (1.8%) — the editor mostly *maintains* rather than *expands*. **(c)** Per-participant size before vs. after editing; median reduction $\approx 18\times$, minimum $9\times$. Per-PID counts and editor ablations are in Appendices A.3, E.1.

4.3 MEMORA-Bench: An Evaluation Suite for Embodied Action Memory

We evaluate embodied action memory formed from continuous embodied experience along two complementary dimensions: *faithfulness to past experience* and *support for future action* — a combination not jointly evaluated by these benchmark families (Fu et al., 2025; Mangalam et al., 2023; Maharana et al., 2024; Wu et al., 2025). MEMORA-Bench therefore pairs two interfaces over the same participant-specific memory state, formed from 45 hours of EPIC-KITCHENS extension video across 18 participants. **MEMORA-Embodied Memory Assessment** (**EAM-QA**) probes retrospective faithfulness with multiple-choice questions over four EAM types (SPREF: preferences; SHABIT: habits; SROUTINE: ordered procedures; ERECALL: within-video recall). It includes answerable items and cross-participant E-correct controls, all with an explicit *information not available* option E to separate missing target-participant evidence from forced guessing. **MEMORA-Planning** probes prospective utility by producing grounded plans from participant-specific goals, split into **Replay** (observed workflows) and **Generalize** (transfer, composition, or fully novel goals). Both interfaces are automatically constructed without inspecting any memory system's output, with ground truth derived from the participant's recordings rather than MEMORA's memory. Released benchmark items are static artifacts and the MEMORA pipeline itself runs entirely on open-weight models (Appendix E.1), so reproducing the headline results does not require closed-weight LLM access. Construction and quality-control details are

in Appendices B and C.5; dataset, conditions, and metrics follow in §5.1.

5 Experiments

5.1 Experimental Setup

Evaluation logic. The experiments separate three questions: whether online editing improves the memory written from perception, whether offline consolidation adds reusable cross-event knowledge, and whether the resulting complete memory interfaces support retrospective assessment and prospective planning. Controlled raw/processed and episodic/full pairs address the first two questions; cross-representation comparisons test the system-level value of each representation together with its read interface, and the robot study checks whether the resulting language plans can support physical execution.

Data and protocols. We evaluate 45 hours from the EPIC-KITCHENS-100 extension (Damen et al., 2022), comprising 201 egocentric videos from all 18 participants for whom consolidated **Inferred Knowledge** memory banks are built (8–15 sessions each). The recordings span participant-specific kitchens, objects, and recurring routines; memory construction and evaluation remain participant-scoped throughout. **EAM-QA** contains N=2,763 multiple-choice items—2,212 answerable questions and 551 E-correct controls—with four content options A–D plus an explicit information-not-available option E, across the four EAM types defined in §4.3. **MEMORA-Planning** contains 207 **Replay** (in-distribution) and 153 **Generalize** (out-of-distribution) tasks; tier-stratified subsets are reported in Appendix C.5.

Conditions and metrics. We compare seven conditions organized into controlled within-family contrasts: **Parametric** (No Memory), **Flat-1D**, a Socratic-style chronological text interface (Zeng et al., 2022), and **Graph-2D**, an entity-relation memory adapted from M3-Agent (Long et al., 2025), in both *raw* and MEMORA-edited (*processed*) variants, **MEMORA-Episodic** (no offline consolidation), and full **MEMORA**. All conditions fix the perception stream, answer backbone, and agent budget; each representation uses its intended read interface. Raw/processed pairs within Flat-1D and Graph-2D, and the MEMORA-Episodic/full pair, keep that interface fixed and isolate online editing or offline consolidation. Tasks are generated by pipelines that never inspect memory-system outputs. EAM-QA reports a conditional diagnostic on questions missed by Parametric for which MEMORA selects a content option A–D rather than option E; all conditions are evaluated on this same subset. This operational filter does not condition on MEMORA being correct and is reported alongside, not instead of, full-benchmark accuracy and E-selection behavior in Appendix B.5. Planning keeps Parametric as a real prior baseline and reports **Robot-Grounded Plan score (RGP)**, the mean of the condition-level **OrderExec**, **KeyObj**, and **PrefAdh** scores; auxiliary planning metrics are in Appendices C and C.6. Unless noted, perception and editing use Qwen2.5-Omni-7B and Qwen3-30B-A3B-Instruct-2507; the four answer backbones and implementation details are listed in Appendix F.

5.2 MEMORA-Embodied Memory Assessment Results

MEMORA improves both cross-video and single-video memory probes. Table 3 reports per-type accuracy under the parametric-missed, MEMORA-committed diagnostic; Appendix B.5 reports the full benchmark and both conditional views. Under this diagnostic view, MEMORA is best on all four question types and overall, leading the strongest non-MEMORA condition (FLAT-1D after online editing) by +20.5 points on Gemma-4-31B-it; the lead holds on both the cross-video semantic probes (SPREF/SHABIT/SROUTINE) and the single-video episodic probe (ERECALL), showing gains on both cross-session semantic probes and within-video episodic recall.

Both online editing and offline consolidation contribute. Online editing improves Graph-2D on all four question types (+7.0 points overall), whereas its effect on Flat-1D is smaller and mixed (+2.0 overall, with a slight drop on SPREF). Even without consolidation, MEMORA-Episodic is particularly strong on within-video ERECALL (73.8), indicating that typed episodic organization already yields strong within-video recall. Offline consolidation adds +14.4 points over MEMORA-Episodic, with its largest gain on SROUTINE (+24.6), followed by SHABIT (+13.7), SPREF (+9.4), and ERECALL (+6.0). This pattern is consistent with consolidation contributing more to cross-event than within-video memory; additional store/editing and consolidation ablations are reported in Appendix B.

Table 2: **From memory formation to planning use.** Three cases link formed memory to planning; full traces are in Appendix Table 14.

Capability tested	Real MEMORA evidence/output	Why it matters
Track one object through state changes. A dirty plate appears at 40 s, 50 s, and 540 s.	**Entity Memory:** `plate_with_food_residue`; `idle@countertop` → `being_used@countertop`, preserving identity and state history.	Planning needs the current state rather than duplicate sightings; editing reduces entity records by median ≈ 18× (Figure 2).
Ground remembered evidence in action. Goal: help P03 clean the washing-up bowl.	**Typed retrieval:** retrieve "wash bowl"; ground the black bowl, green sponge, chrome faucet, and metal drying rack. **Plan:** wash with the sponge under the faucet, then place on the rack.	The plan binds a reusable routine to participant-specific objects and places; No Memory and Flat-1D retrieve generic or mismatched evidence.
Consolidate repeated experience. SPREF: does P04 prefer a spatula or fork for eggs? Gold: spatula.	**Evidence:** P04 repeatedly uses a wooden spatula for yellow mixtures and eggs (P04_116, turns 28 and 39). **Answer:** spatula.	Repeated evidence captures a stable preference; No Memory, Flat-1D, and Graph-2D return information unavailable.

Table 3: **MEMORA-Embodied Memory Assessment per-type accuracy (%) on Gemma-4-31B-it** (N=1,122 EAM-QA questions under the parametric-missed, MEMORA-committed diagnostic over the full 18 participants). SPREF/SHABIT/SROUTINE aggregate evidence across sessions; ERECALL is within a single video. Δ is MEMORA minus MEMORA-Episodic; **bold** marks the best per column. Parametric is omitted because this view focuses on questions it answered incorrectly; see Appendix B.5. Per-backbone reproductions are in Appendix Table 10.

	Cross-video			Single-video	
Memory condition	**SPREF**	**SHABIT**	**SROUTINE**	**ERECALL**	**Overall**
Baselines built directly from the perception stream					
Flat-1D (Zeng et al., 2022) (raw)	63.2	36.8	53.8	51.2	52.0
Graph-2D (Long et al., 2025) (raw)	50.9	21.4	37.1	55.4	40.0
Baselines rebuilt after MEMORA's online editing					
Flat-1D (Zeng et al., 2022) (processed)	62.4	41.4	56.8	53.0	54.0
Graph-2D (Long et al., 2025) (processed)	57.1	27.0	48.6	57.1	47.0
MEMORA memory stores					
MEMORA-Episodic (no consol.)	67.4	47.0	56.8	73.8	60.1
MEMORA (full)	**76.8**	**60.7**	**81.5**	**79.8**	**74.5**
Δ from consolidation (points)	**+9.4**	**+13.7**	**+24.6**	**+6.0**	**+14.4**

Cross-backbone consistency. Across the other three open-weight backbones under the same parametric-missed, MEMORA-committed diagnostic, MEMORA leads the strongest non-MEMORA condition on every question type, and the offline-consolidation lift remains positive for every type and backbone. Per-backbone tables and complementary evaluation views are in Appendices B.6 and B.5.

5.3 MEMORA-Planning Results

Planning tests whether formed memory can guide future action rather than only answer retrospective questions. Replay tasks contain a previously observed workflow, while Generalize tasks require transfer or composition when no single stored episode matches the goal.

MEMORA's strongest planning gains occur when exact replay is unavailable. Across the four answer backbones, a MEMORA-family variant is best in all eight (backbone, benchmark) cells. For three of the four backbones, the gain is larger on Generalize than on Replay, suggesting that MEMORA is especially useful when the model composes beyond a matching observed episode. In the Gemma-4-31B-it case, the contrast is informative: MEMORA-Episodic is strongest on Replay, where a matched workflow can suffice, whereas full MEMORA is strongest on Generalize, where consolidated knowledge supports composition. All non-MEMORA retrieval baselines fall below the no-memory prior on Generalize, while MEMORA improves RGP by +16.4% over the strongest non-MEMORA baseline, suggesting that access to past experience alone is insufficient without an interface

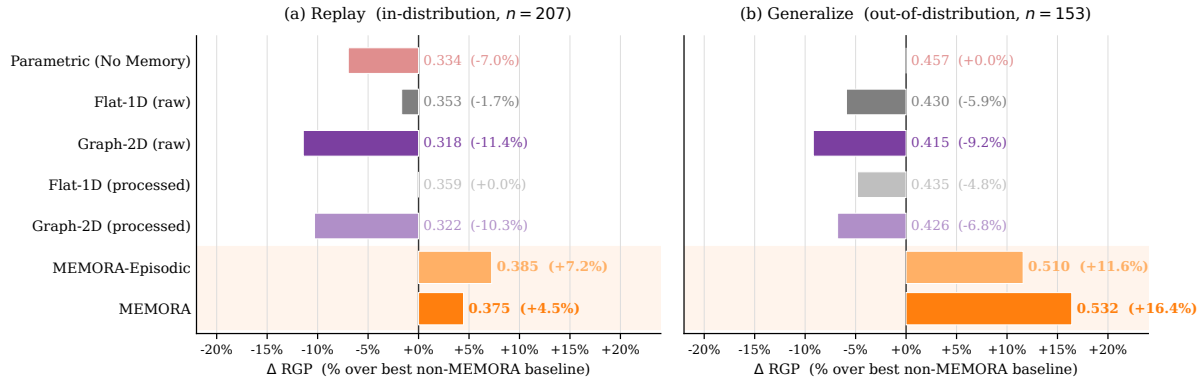


Figure 3: **RGP on Gemma-4-31B-it** (highlighted Generalize case). Bars show absolute RGP and $\Delta\%$ vs. the strongest non-MEMORA baseline (0% line). **(a)** Replay (n=207): MEMORA-Episodic $+7.2\%$. **(b)** Generalize (n=153): both MEMORA variants exceed the parametric prior; full MEMORA reaches $+16.4\%$, while non-MEMORA retrieval baselines fall below no memory. Cross-backbone and per-axis details are in Appendix C.

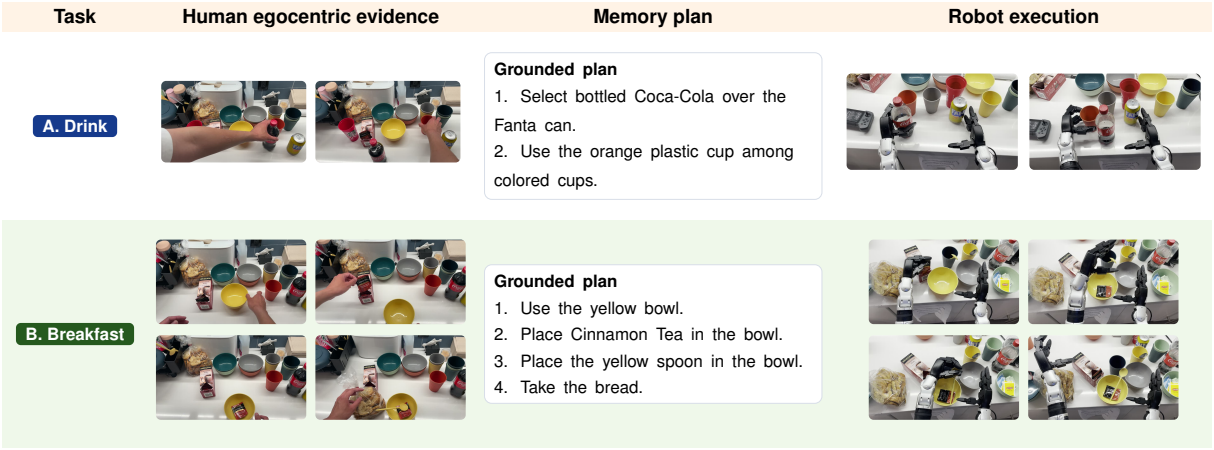


Figure 4: **Physical robot demonstration.** MEMORA transfers memory formed from human egocentric video to a rule-based Unitree G1 for *Prepare Drink* and *Breakfast*, grounding preferences, objects, and sequence.

that exposes useful evidence for transfer. This is the planning behavior illustrated in Table 2: reusable procedure is retrieved first and then grounded in remembered entities and places. Per-cell results and cross-backbone scaling are in Appendix C (Figure 15).

Offline consolidation benefits stronger planners on Generalize. On Replay, MEMORA-Episodic often matches or exceeds full MEMORA, consistent with the relevant routine already being available as episodic evidence. On Generalize, full MEMORA has higher RGP than MEMORA-Episodic on the two strongest reasoning backbones (Q35 $+.025$; G31 $+.022$), while smaller backbones are neutral or negative under the same comparison. On Gemma-4-31B-it, per-axis tables show the largest relative gains on procedure order ($+43\%$) and preference adherence ($+14\%$), while key-object coverage remains competitive with the parametric prior (Appendices C, C.6).

Tool use and ablations provide supporting evidence. Flat-1D uses keyword `search`, whereas MEMORA provides typed access to routines, objects, and preferences (Appendix C; Fig. 16). Replacing the main editor with Qwen3-14B and Omni perception with VL-only perception reduces EAM-QA by 5.8 and 2.7 points, respectively; removing **Inferred Knowledge** shifts 35% of planning calls toward episodic activity search. The editor ablation also raises E-selection from 42.8% to 62.3%, consistent with reduced usability rather than only reduced completeness (Appendices E.3, E.1). Additional breakdowns, grounding metrics, and worked examples are in Appendix C.

5.4 Physical Robot Execution

We include a physical demonstration to test whether memory formed from human egocentric experience can guide a robot acting in the same scene. Its participant-specific language plans are executed on a Unitree G1 by a fixed rule-based controller, isolating grounding without training on robot demonstrations. Baseline plans often select wrong objects, miss remembered preferences, or produce incomplete sequences; with MEMORA plans, the robot succeeds on both tasks. We treat this as a qualitative execution check, not a standalone robot benchmark; Figure 4 summarizes both tasks, with full traces and protocol in Appendix D.

6 Discussion

Memory as a substrate for reasoning and planning. MEMORA separates reusable procedural evidence from participant-specific grounding. This separation is most consequential on GENERALIZE, where no single stored episode fully specifies a new goal and planning depends on composing the two. REPLAY can often recover a matched episodic workflow, whereas GENERALIZE asks the planner to compose knowledge accumulated across prior experience and reinstantiate that knowledge with remembered entities, states, and locations.

What the ablations isolate. Within-family contrasts isolate online editing and offline consolidation by holding the read interface fixed; cross-family comparisons instead evaluate complete memory interfaces and are best interpreted at the system level because representation, organization, and access change together.

Why memory can hurt. The Generalize results show that memory presence alone is insufficient: flat retrieval can fall below the parametric prior when large, weakly structured passages introduce irrelevant or conflicting evidence. Online editing reduces this context pollution, while typed retrieval limits the evidence exposed to the answerer or planner. The baseline pattern is also representation-dependent: Graph-2D is relatively stronger on within-video ERECALL, while Flat-1D is stronger on cross-session semantic probes, suggesting that memory organization changes which kinds of evidence are easiest to access. Offline consolidation contributes a different benefit by converting repeated activity into reusable regularities; its larger gains on the two higher-performing reasoning backbones are consistent with those models making more effective use of the regularities alongside entity and location evidence. Together, these results suggest that memory quality depends on how experience is formed, organized, and exposed at use time, not only on how much history is retained.

Scope within an embodied-agent stack. MEMORA addresses long-horizon context accumulated through a participant's ongoing experience. Vision–language–action policies map instructions and observations toward actions, world or action models reason over possible futures, and low-level controllers execute motions on a particular embodiment. MEMORA instead exposes prior participant-specific places, state changes, procedures, and regularities as semantic–procedural context. Egocentric video is particularly useful here because its longitudinal first-person record captures how a particular person actually uses a familiar environment. Goals such as "prepare a drink" or "prepare breakfast" are under-specified by the current instruction and scene alone: remembered choices of beverage, cup, objects, and order determine what the goal means for that participant. The same egocentric signal can therefore support both action-capability learning and memory formation for later grounding. Closed-loop deployment could further pair this memory-grounded context with an execution policy and a progress monitor that verifies intermediate state transitions and triggers replanning (Skreta et al., 2024; Dai et al., 2026; Liu et al., 2026); Appendix D discusses practical configurations. Accordingly, the Unitree demonstration evaluates grounding for downstream execution, not a new controller or closed-loop robot policy.

7 Conclusion

We introduce **MEMORA**, which forms longitudinal egocentric experience into typed, editable, and consolidated memory, and **MEMORA-Bench** for retrospective faithfulness and prospective planning. Across four backbones, MEMORA achieves the strongest aggregate planning performance among the evaluated complete memory interfaces, with gains up to +16.6% relative RGP on GENERALIZE. A physical-robot demonstration shows participant-specific memory can ground robot plans. Together, these results show that actively formed memory can turn accumulated embodied experience into a persistent basis for future reasoning and action.

Limitations

Domain and horizon. Our evaluation is limited to kitchen activity in the EPIC-KITCHENS-100 extension videos (Damen et al., 2022), with at most 15 sessions per participant. This horizon exposes recurring places, objects, and routines, but not months-long drift, seasonal reorganization, or rare events. Extending MEMORA to other environments and months-to-years horizons may require stronger capacity control, forgetting, and consolidation policies.

Pipeline dependence. MEMORA depends on the quality of perception and write-time memory editing. The ablations in Appendix E.3 show that weaker editors or perception backbones reduce EAM-QA performance, so improving memory formation remains important for both retrieval and planning. Because memory is formed from perceived segments, missed objects can be recovered only if they appear in later evidence, and repeated perception errors may enter consolidated knowledge if subsequent observations do not contradict them.

Evaluation scope. The Flat-1D, Graph-2D, and MEMORA comparisons evaluate complete memory interfaces, including their representations, retrieval tools, and instructions; only the within-family contrasts isolate online editing or offline consolidation. Generalize tasks are proposed and filtered relative to each participant's recorded experience and scored against verified task-level references. They test transfer and composition beyond matched Replay episodes, but do not substitute for an independently authored distribution of external goals.

Physical deployment. The Unitree G1 result in §5.4 is a qualitative transfer check rather than a closed-loop robot benchmark; we do not claim a new controller or learned policy, only that the memory state formed from human egocentric video is usable as language-level grounding for downstream physical action. Privacy, fairness, and dual-use considerations associated with persistent participant-specific memory are discussed in the Ethical Considerations section.

Ethical Considerations

Participant consent and data provenance. MEMORA-Bench is built exclusively on EPIC-KITCHENS-100 (Damen et al., 2022), a publicly released egocentric dataset whose recordings were obtained with participant consent and under institutional ethics review by the original dataset authors. We add no new recordings, no additional identifiers, and no information that was not already present in the source dataset; the participant-specific memories we construct are derived representations of the same released video. Participant identifiers (e.g. `P01`, `P09`) follow the dataset's pseudonymous labels.

Privacy implications of persistent participant-specific memory. A practical embodied agent that maintains months- or years-long memory of an individual's home, routines, preferences, and object choices creates real privacy exposure beyond the present research setting. We frame MEMORA as a research substrate for memory *construction and assessment*, not as a deployment-ready system. Any downstream deployment of similar memory machinery should, at minimum, obtain explicit informed consent (including for downstream uses such as planning or robot transfer), store participant memories under appropriate access control with on-device or end-to-end-encrypted storage where feasible, support per-participant inspection and deletion, and avoid sharing per-participant memories across users. These are engineering and governance prerequisites that lie outside the present paper but which we view as binding conditions for any real deployment.

Population coverage and fairness. EPIC-KITCHENS-100 participants are demographically concentrated and recorded under a single activity protocol; routines, preferences, object inventories, and verbal/non-verbal cues may differ substantially across cultures, household structures, abilities, and economic contexts. We therefore do not claim that MEMORA's behavior generalises to populations outside this protocol, and we treat cross-cultural and cross-ability extensions as both a fairness concern and the cleanest empirical test of the typed-store formulation; we flag this explicitly so that downstream users do not over-generalise from our reported numbers.

Physical deployment. The Unitree G1 demonstration in §5.4 is a qualitative transfer check, not a closed-loop robot benchmark, and uses a rule-based controller rather than a learned visuomotor policy. Real physical deployment of any sys-

tem that converts human egocentric memory into robot behavior introduces safety considerations (workspace design, operator supervision, failure modes under perception error) that are beyond the scope of this paper but which any practical instantiation must address before unsupervised use.

Dual use and intended use. Persistent first-person memory of an individual could in principle be repurposed for surveillance or behavioural monitoring. We explicitly position MEMORA as supporting *participant-aligned* assistive applications — where the memory is constructed for, with the consent of, and for the benefit of the same person whose experience it represents — and not as a third-party monitoring tool. We discourage applications in which a memory of an individual is constructed or queried without that individual's informed, ongoing consent.

Reproducibility and openness. The MEMORA pipeline uses only publicly released open-weight language and multimodal models (full list and identifiers in Appendix F); we do not depend on closed-weight APIs for any headline result. This is intended both to support independent reproduction and to keep the cost of replicating, scrutinising, or auditing the system within reach of academic groups.

Acknowledgments

For this work, Xiu Yuan and Chongjie Zhang were supported by AF Office of Scientific Research (AFOSR) under grant FA9550-25-1-0318.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, and 26 others. 2022. Do as I can, not as I say: Grounding language in robotic affordances. *Preprint*, arXiv:2204.01691.

John R. Anderson. 1983. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA.

Richard C. Atkinson and Richard M. Shiffrin. 1968. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet T. Spence, editors, *Psychology of Learning and Motivation*, volume 2, pages 89–195. Academic Press, New York.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, and 5 others. 2026. π_0: A vision-language-action flow model for general robot control. *Preprint*, arXiv:2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, and 35 others. 2023. RT-2: Vision-language-action models transfer web knowledge to robotic control. *Preprint*, arXiv:2307.15818.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *Preprint*, arXiv:2504.19413.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2023. Diffusion Policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*.

Tingjun Dai, Mingfei Han, Tingwen Du, Zhiheng Liu, Zihao Zhang, Zhihui Li, Salman Khan, Jun Yu, and Xiaojun Chang. 2026. See, Plan, Rewind: Progress-aware Vision-Language-Action models for robust robotic manipulation. *Preprint*, arXiv:2603.09292.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2022. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55.

Pengfei Du. 2026. Memory for autonomous LLM agents: Mechanisms, evaluation, and emerging frontiers. *Preprint*, arXiv:2603.07670.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *Preprint*, arXiv:2405.21075.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. Ego4D: Around the world in 3,000 hours of egocentric video. *Preprint*, arXiv:2110.07058.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner monologue: Embodied reasoning through planning with language models. *Preprint*, arXiv:2207.05608.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. 2024. EgoMimic: Scaling imitation learning via egocentric video. *Preprint*, arXiv:2410.24221.

Taeyoon Kwon, Dongwook Choi, Hyojun Kim, Sunghwan Kim, Seungjun Moon, Beong-woo Kwak, Kuan-Hao Huang, and Jinyoung Yeo. 2026. Embodied agents meet personalization: Investigating challenges and solutions through the lens of memory utilization. *Preprint*, arXiv:2505.16348.

John E. Laird. 2012. *The Soar Cognitive Architecture*. MIT Press, Cambridge, MA.

Mingcong Lei, Honghao Cai, Yuyuan Yang, Yimou Wu, Jinke Ren, Zezhou Cui, Liangchen Tan, Junkun Hong, Gehan Hu, Shuangyu Zhu, Shaohan Jiang, Ge Wang, Junyuan Tan, Zhenglin Wan, Zheng Li, Zhen Li, Shuguang Cui, Yiming Zhao, and Yatong Han. 2026. RoboMemory: A brain-inspired multi-memory agentic framework for interactive environmental learning in physical embodied systems. *Preprint*, arXiv:2508.01415.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Preprint*, arXiv:2005.11401.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024. MVBench: A comprehensive multi-modal video understanding benchmark. *Preprint*, arXiv:2311.17005.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. *Preprint*, arXiv:2209.07753.

Yushan Liu, Peibo Sun, Xintao Chao, Zhenyang Yang, Yifan Xie, Lingfeng Zhang, Shoujie Li, Chenyu Tang, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. 2026. CheckVLA: Execution-time verification with action-conditioned world model for long-horizon mobile manipulation. *Preprint*, arXiv:2607.26789.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. 2025. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *Preprint*, arXiv:2508.09736.

Xiaoyu Ma, Lianyu Hu, Wenbing Tang, Zixuan Hu, Zeqin Liao, Zhizhen Wu, and Yang Liu. 2026. BrainMem: Brain-inspired evolving memory for embodied agent task planning. *Preprint*, arXiv:2604.16331.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. *Preprint*, arXiv:2402.17753.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *Preprint*, arXiv:2308.09126.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457.

Pearl Mody, Mihir Panchal, Rishit Kar, Kiran Bhowmick, and Ruhina Karani. 2026. CraniMem: Cranial inspired gated and bounded memory for agentic systems. *Preprint*, arXiv:2603.15642.

Karim Nader. 2003. Memory traces unbound. *Trends in Neurosciences*, 26(2):65–72.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. MemGPT: Towards LLMs as operating systems. *Preprint*, arXiv:2310.08560.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. TEACh: Task-driven embodied agents that chat. *Preprint*, arXiv:2110.00534.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *Preprint*, arXiv:2304.03442.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Cheluva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, and 20 others. 2026. EgoVerse: An egocentric human dataset for robot learning from around the world. *Preprint*, arXiv:2604.07607.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. *Preprint*, arXiv:1912.01734.

Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. 2024. RePLan: Robotic replanning with perception and language models. *Preprint*, arXiv:2401.04157.

Robert Stickgold. 2005. Sleep-dependent memory consolidation. *Nature*, 437(7063):1272–1278.

Endel Tulving. 1972. Episodic and semantic memory. In Endel Tulving and Wayne Donaldson, editors, *Organization of Memory*, pages 381–403. Academic Press, New York.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. *Preprint*, arXiv:1410.3916.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. LongMemEval: Benchmarking chat assistants on long-term interactive memory. *Preprint*, arXiv:2410.10813.

Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. 2026. RL Token: Bootstrapping online RL with Vision-Language-Action models. *Preprint*, arXiv:2604.23073.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2026. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *Preprint*, arXiv:2508.19828.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 2025. 3D-Mem: 3D scene memory for embodied exploration and reasoning. *Preprint*, arXiv:2411.17735.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Chongrui Ye, Yuxiang Liu, Yu Wang, Haofei Yu, Yining Zhao, Ge Liu, Julian McAuley, and Jiaxuan You. 2026. Auto-Dreamer: Learning offline memory consolidation for language agents. *Preprint*, arXiv:2605.20616.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. 2025. Policy Decorator: Model-agnostic online refinement for large policy model. In *International Conference on Learning Representations*.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *Preprint*, arXiv:2204.00598.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *Preprint*, arXiv:2404.13501.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. 2026a. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *Preprint*, arXiv:2602.16710.

Xinyi Zheng, Yunze Liu, Chi-Hao Wu, Fan Zhang, Hao Zheng, Wenqi Zhou, Walterio W. Mayol-Cuevas, and Junxiao Shen. 2026b. SpatialMem: Metric-aligned long-horizon video memory for language grounding and QA. *Preprint*, arXiv:2601.14895.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. MemoryBank: Enhancing large language models with long-term memory. *Preprint*, arXiv:2305.10250.

Appendix roadmap. The appendix is organized around construction, evaluation, and controlled analysis. Appendix A specifies the memory-bank construction pipeline from egocentric video. Appendix B defines the retrospective Embodied Memory assessment, participant scope, and unsupported-answer controls. Appendix C specifies the MEMORA-Planning benchmark and rule-based evaluation panel. Appendix D documents the physical robot demonstration used as a qualitative deployment check. Appendix E defines controlled ablations over processor and answer-agent choices. Appendix F consolidates the model registry, inference stack, perception preprocessing, and retrieval/agent-loop settings shared across all of the above.

A Memory-Bank Construction Pipeline

Appendix A specifies how MEMORA constructs the typed memory bank used by the Embodied Memory and Planning evaluations. The pipeline maps egocentric EPIC-KITCHENS videos into the four stores defined in Section 4: **Environment Memory**, **Entity Memory**, **Activity Memory**, and offline **Inferred Knowledge**. The presentation follows processor roles rather than engineering stage names.

Terminology (Inferred Knowledge and MEMORA-Episodic). In the main paper, cross-episode routines, habits, and preferences are typed as **Inferred Knowledge** and populated by offline consolidation (Table 1). This appendix decomposes that offline path into two processors: an **Offline Consolidation Processor** (per-video storage preferences, organizational habits, workflow patterns, and repeated action sequences) and an **Inferred Knowledge Enrichment Processor** (participant-level routine-skill indices, generated preferences, and retrieval text). Together they form the consolidated **Inferred Knowledge** bank used at evaluation time. Full **MEMORA** loads the enriched participant memory bank: EAM-QA reaches consolidation outputs through `search_patterns` and, under full MEMORA, also indexes enriched routine skills and generated preferences in the same unified retrieval interface; Planning additionally advertises dedicated `get_routine_skill` and `get_preferences` entry points over the enriched consolidated **Inferred Knowledge** slice. The label **MEMORA-Episodic** matches the main-paper ablation on online **Environment Memory**/**Entity Memory**/**Activity Memory**: EAM-QA evaluates the per-video memory bank before enrichment through the standard layered search tools; Planning evaluates the same episodic stores through `search_objects`, `get_object_history`, and `search_activities`.

A.1 Implementation Overview

Each video is divided into non-overlapping 10-second turns. In implementation terms, the *Perception Processor* realizes the Segment Encoder ϕ, while the *Online Memory Processor* realizes the write-time EDIT update described in Section 4. For turn t, the Perception Processor emits a layered observation $z_t = (z_t^{\text{env}}, z_t^{\text{act}}, z_t^{\text{ent}})$. The Online Memory Processor then updates a persistent memory state $\mathcal{M}_t = (\mathcal{M}_t^{\text{env}}, \mathcal{M}_t^{\text{act}}, \mathcal{M}_t^{\text{ent}}, \mathcal{M}^{\text{inf}})$ in timestamp order:

$$z_t = \phi(v_t, c_{t-1}), \qquad \mathcal{M}_t = \textsc{Edit}(\mathcal{M}_{t-1}, z_t). \tag{3}$$

In this notation, v_t is the 10-second clip and c_{t-1} is a bounded textual summary of earlier turns in the same video. Previous video frames are not retained in the model KV cache across turns.

Each participant memory bank comprises four typed stores: turn-local perception records, per-video environment and activity logs, an edited entity registry with operation history, and a participant-level consolidated **Inferred Knowledge** slice.

A.2 Memory Update Semantics

The Perception Processor produces turn-local observations only; it does not reconcile duplicate entities or maintain long-horizon state. Cross-turn memory formation is handled by the Online Memory Processor. Activity memory is append-only because it represents the event stream. Environment memory is merged by stable location identifiers and retains a history of updates. Entity memory is edited by the online memory processor: for each incoming object observation, the model emits one of ADD, UPDATE, DELETE, or NOOP against the compact current registry. When the entity registry is too large for the prompt budget, dense retrieval (Appendix F) selects the most relevant existing entities for the edit decision.

The Offline Consolidation Processor reads completed memories for a participant and writes per-video consolidation summaries (storage prefer-

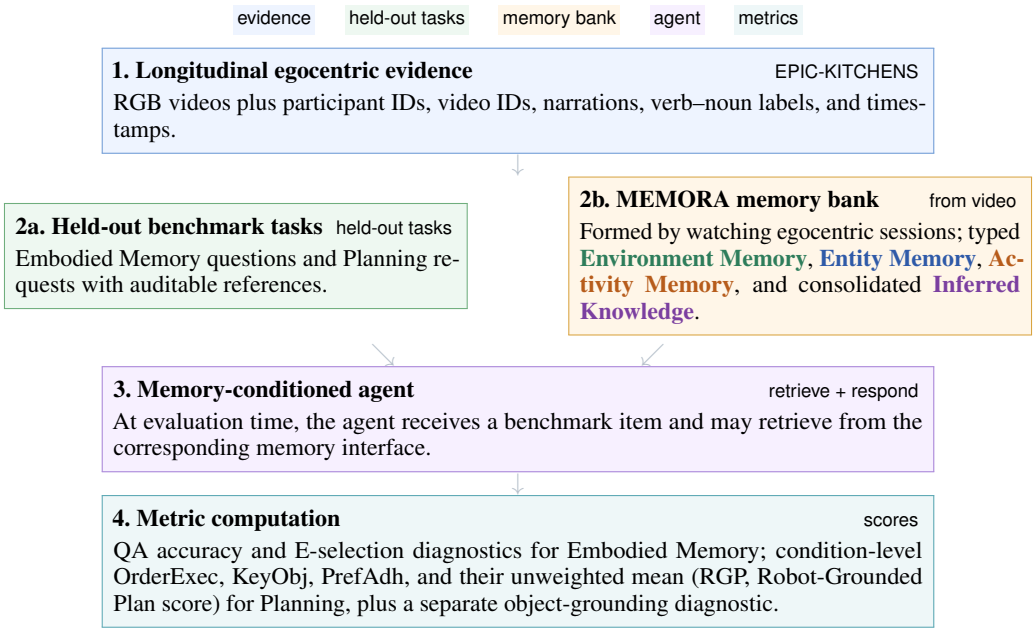


Figure 5: **Appendix visual roadmap for MEMORA-Bench.** The appendix separates four concepts that are easy to conflate: longitudinal evidence, held-out benchmark tasks, memory-bank construction from video, and memory-conditioned evaluation. Subsequent figures expand the memory-bank construction pipeline (Figure 6), the Embodied Memory benchmark generator (Figure 10), the Planning benchmark generator (Figure 13), and the final retrieval-and-scoring loop (Figure 9).

ences, habits, workflow patterns, and action sequences). These summaries support preference, habit, and routine reasoning; in the released retrospective QA panel, they are exposed only within the complete fixed memory bank of the question's target participant. The **Inferred Knowledge** Enrichment Processor subsequently derives planning-oriented memory from the participant slice. It creates routine-skill structures and retrieval text deterministically, then uses the main-paper enrichment LLM in Table 23 to produce evidence-grounded generated preferences; a closed-weight alternative is reported as a sanity check in Appendix E.1.

A.3 Editor Activity and Memory Compression

This subsection documents the data behind Figure 2 in Method §4.1. The 18-participant figure traces the cumulative size of Entity Memory under two write policies on the same main-paper perception stream and online memory processor (Table 23): a hypothetical *without-editing* policy that would persist one record per entity observation (every `state_history` event the perception layer emits with a turn-aligned timestamp), and the MEMORA policy that emits one record per unique entity and folds repeat encounters into its `state_history` list. The compression ratio reported per participant is the ratio of these two record counts at end-of-session; segment-level cumulative curves use turn-aligned events only (events without a resolved `turn_id` are dropped from both numerator and denominator so they cannot inflate either side). Panel (b)'s editor decision distribution is read off the released editor operation logs; these decision counts are fully available for the P01–P04 calibration cohort and partially available for the 18-participant evaluation cohort, so the distribution panel is reported on P01–P04 alone while panels (a) and (c) use the full 18-participant cohort.

Per-participant size and compression. Figure 2, panels (a) and (c), report the per-participant figures. Every participant achieves at least a $9\times$ reduction; the typical (median) participant achieves $18.1\times$, and three participants exceed $30\times$. PIDs with longer sessions and higher entity reuse (P12, P23, P25, P35) reach the largest ratios because the editor turns repeated encounters with the same kitchen objects into state updates rather than new records; participants with shorter sessions (P09) sit at the floor of the range.

Editor decision distribution (P01–P04). The calibration-cohort editor logs (P01–P04) report the editor's decisions over n=37,648 object observations: ADD = 2,995 (8.0%), UPDATE = 16,718

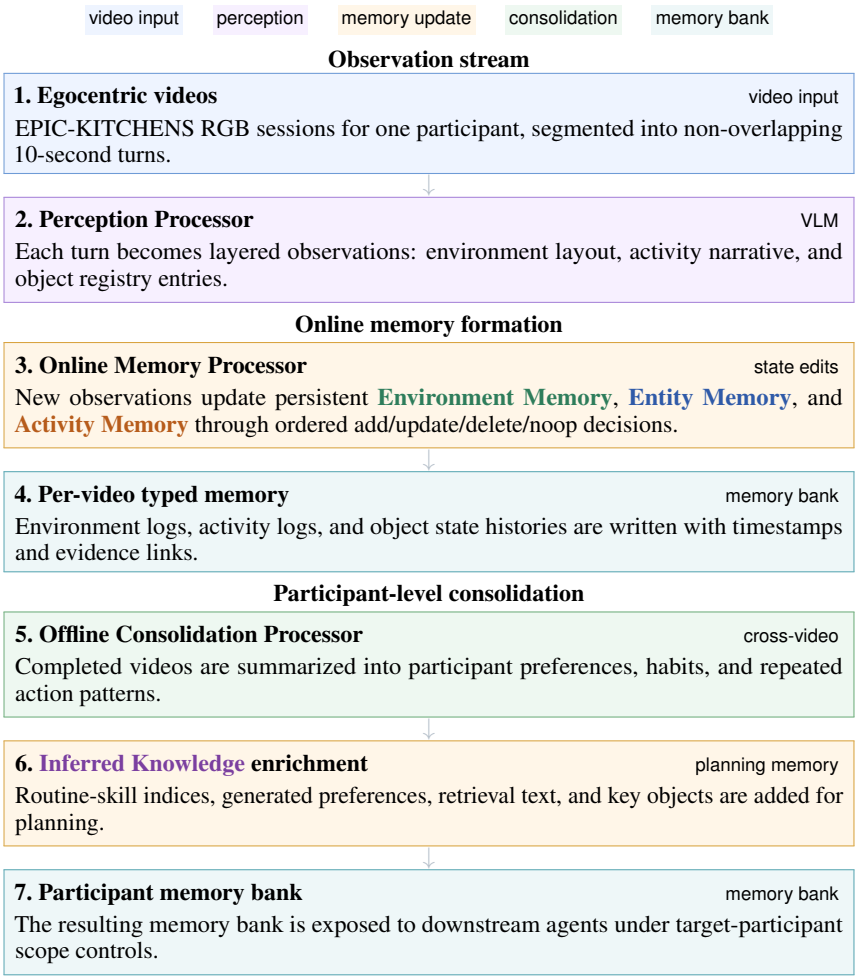


Figure 6: **MEMORA memory-bank construction pipeline.** The system watches egocentric videos turn by turn, converts each clip into structured observations, updates typed memory online, and then consolidates participant-level routines and preferences for downstream Embodied Memory and Planning evaluations.

(44.4%), DELETE = 691 (1.8%), NOOP = 17,244 (45.8%). Two readings are worth recording. First, NOOP + UPDATE together account for 90.2% of decisions: most of the editor's work is recognizing that an entity is already in memory and either revising its state or letting it stand, not adding new records. Second, DELETE is rare (1.8%) but never zero — the editor does retract identifications that subsequent observations disconfirm, which a frozen, append-only memory could not. This decision profile is the per-event mechanism behind the 18.1× aggregate compression in Figure 2: every NOOP on a previously-seen object is one entity observation that does not become a new record, and every UPDATE is one observation that updates an existing record rather than creating one.

A.4 Processor Prompt Templates

The construction pipeline in Figure 6 uses fixed prompt templates with runtime variables filled from the current segment, current memory state, or participant memory slice. Figure 7 summarizes the four prompted processors, and Figure 8 gives compact records for the four MEMORA memory types. Deterministic rule components in the **Inferred Knowledge** Enrichment Processor have no prompt.

A.5 Memory Type Examples

Figure 8 provides abbreviated records for the four memory types used by MEMORA: **Environment Memory**, **Entity Memory**, **Activity Memory**, and **Inferred Knowledge**. Field names are preserved from the released records; values are shortened to make the representation readable in print.

Table 4: Main-paper memory-bank construction processors. The table specifies processor contracts (input scope, outputs, and construction notes); model identifiers appear in Table 23 and temporal preprocessing in Appendix F.

Processor	Input scope	Output	Construction notes
Perception Processor	One 10-second egocentric clip	Turn-local `environment`, `activity_narrative`, and `object_registry` records	Run independently per turn; 2 FPS, 24-frame cap, audio on (Omni's built-in audio path); vision-only ablation drops audio at decode time.
Online Memory Processor	Ordered turns for one video	Per-video `environment_log`, `activity_log`, and edited `object_registry`	Per-video scope; dense retrieval pre-filter when the entity registry exceeds the prompt budget.
Offline Consolidation Processor	Completed videos for one participant	Per-video consolidation summaries: storage preferences, organizational habits, workflow patterns, and action sequences	Post-video offline pass; reuses the loaded online memory model.
Inferred Knowledge Enrichment Processor	One participant memory slice	Routine-skill indices, retrieval text, key objects, and evidence-grounded generated preferences	Rule skeleton plus one LLM preference pass per participant; closed-weight alternative in Appendix E.1.

A. Processor prompt templates

Perception Processor

```
Input: current 10-second clip; optional previous context
and detected objects.
Instruction: analyze visible egocentric kitchen activity,
use relative timestamps in [0,10]s, reuse object
identifiers when available.
Output JSON: environment layout/zones/relations; activity
summary and action_breakdown; object_registry with visual
properties, spatial info, and state.
```

Online Memory Processor

```
Input: compact current object_registry plus full new object
states.
Instruction: edit Entity Memory with ADD, UPDATE, DELETE,
or NOOP; preserve movement_trajectory; keep exact object_id
unless adding a new object.
Output JSON: object_operations with event, object_id,
changes/data, and reason.
```

Offline Consolidation Processor

```
Input: participant_id, completed object_registry, and
activity_log.
Instruction: infer only from completed memories; identify
storage preferences, workflow preferences, organizational
habits, and recurring task patterns.
Output JSON: storage_preferences, organizational_habits,
workflow_patterns.
```

Inferred Knowledge Enrichment Processor

```
Input: participant, max_skills, allowed_subtypes, and
evidence_items.
Instruction: convert low-level evidence into reusable
participant-level skill memories; use only supplied
evidence IDs; prefer concrete object/action/context skills.
Output JSON: generated_preferences with subtype, keywords,
confidence, and supporting_evidence_ids.
```

Figure 7: **Prompt templates for MEMORA memory-bank construction.** Each box preserves the operational contract of a prompted processor: what enters the module, what instruction constrains it, and what structured fields it must emit.

Environment Memory record

```
environment_log: [
  location_id: sink_area
  current_state: layout_description, zones=[sink_basin,
drying_rack, prep_counter], spatial_relations=[drying_rack
RIGHT_OF sink_basin]
  history: turn_id + update
]
```

Activity Memory record

```
activity_log: [
  turn_id: 12; time_window: 120.0-130.0
  summary: participant washes a knife with a sponge in the
sink
  action_breakdown: [0-3s scrubs knife_metal; 3-6s rinses
knife_metal]
]
```

Entity Memory record

```
object_registry.knife_metal:
  visual_properties: material=metal, size=medium
  spatial_info: location=in sink, zone=sink_basin
  state: current_state=being washed, held_by=right_hand
  state_history: turn 11 on counter/idle; turn 12 in
sink/being washed
```

Inferred Knowledge record

```
Consolidation (per-video): workflow pattern "dirty utensils
are washed at the sink before reuse"; confidence=0.82
Enrichment (participant-level): generated preference that
participant often uses a spatula to stir potatoes or
vegetables; keywords=[spatula, stir, potatoes, vegetables];
supporting_evidence_ids=[E001,E005]
```

Figure 8: **Abbreviated JSON-style examples of MEMORA memory records.** The panel keeps literal schema field names but compresses values so the four record types can be compared without a split listing block.

A.6 Memory Stores and Evaluation Scope

Evaluation scope. In retrospective Embodied Memory, each question is evaluated against only the fixed memory bank of its target participant. Answerable items are derived from that participant's observed corpus; E-correct controls are remapped from a different donor participant and therefore lack supporting target-memory evidence. The released headline panel has no per-question temporal anchor and does not apply state rollback. In Planning, scope control restricts both benchmark tasks and retrieval to videos already present in the participant's consolidated memory slice, so the agent is not evaluated on episodes it never encoded. Table 5 summarizes these controls for each memory store.

A.7 Limitations

The memory bank inherits errors from the perception stream: missed objects, weak audio cues, and ambiguous hand-object interactions can propagate into later processors. The Online Memory Processor can repair cross-turn state, but visually similar objects may still be duplicated or stale attributes may persist when evidence is weak. The released retrospective panel evaluates a participant's completed memory bank and therefore does not test within-video prefix recall; a future temporally anchored variant would require explicit memory slicing or rollback. Finally, consolidated **Inferred Knowledge** is optimized for planning and preference retrieval; it should not be interpreted as an exhaustive transcript of every observed event.

Table 5: Primary memory stores and evaluation scope. The final column distinguishes construction-time locality from the participant-level scope used by each evaluation.

Store	Producer	Content	Evaluation scope
Per-segment perception record	Perception Processor	Turn-local `environment`, `activity`, and `object` observations	Generated from the current 10-second segment plus bounded text context only
`environment_log`, `activity_log`	Online Memory Processor	Timestamped spatial and activity memories	Restricted to the target participant's fixed memory bank
`object_registry`	Online Memory Processor	Edited entity memory with add/update/delete/noop decisions and state history	Full target-participant registry is available in headline EAM-QA
Consolidation summaries	Offline Consolidation Processor	Per-video storage preferences, organizational habits, workflow patterns, and action sequences	Included according to the evaluated episodic/full condition
Consolidated **Inferred Knowledge** slice	**Inferred Knowledge** Enrichment Processor	Routine skills, generated preferences, retrieval text, and key objects	Planning retrieval limited to source videos in the participant memory slice

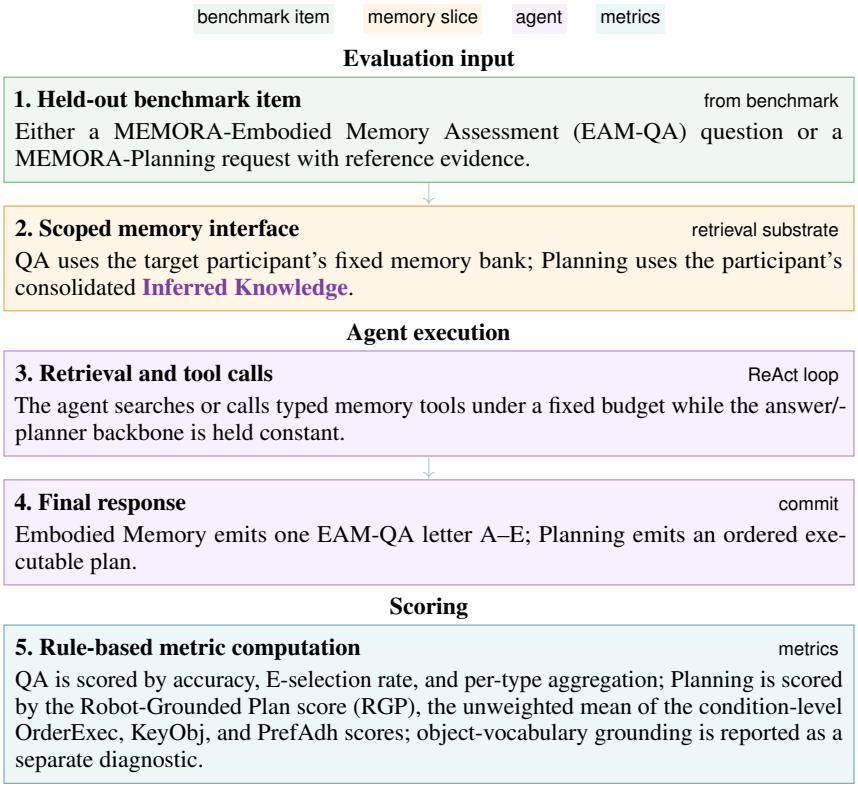


Figure 9: **Memory-conditioned evaluation and scoring loop.** The benchmark item and the scoped memory interface are kept separate: benchmark tasks define what should be answered, while memory substrates define what the agent can retrieve before committing to an answer or plan.

B MEMORA-Embodied Memory Assessment

Appendix B defines the retrospective QA arm of MEMORA-Bench. The **MEMORA-Embodied Memory Assessment** measures whether participant-specific memory is faithful, accessible, and correctly scoped, including when the target participant's memory lacks supporting evidence. The annotation-grounded release supports the headline four-way split (ERECALL, SPREF, SHABIT, SROUTINE) on 45 hours of EPIC-KITCHENS-100 (Damen et al., 2022) extension video across 18 participants. Generation is architecture-agnostic: questions are mined from action annotations and do not presuppose MEMORA's four-substrate layout. §B.1 specifies benchmark construction and question grounding; §B.4 specifies the evaluation protocol. The EAM-QA setup evaluates retrieval from the constructed memory bank rather than generic video question answering.

B.1 Benchmark Construction

EPIC-KITCHENS inputs. The construction builds on **EPIC-KITCHENS-100** (Damen et al., 2022), an egocentric kitchen corpus in which each **participant** records multiple **videos** in the same home over time. The public release supplies:

- **RGB video** for every session (*not used in generation*; required at evaluation by the perception front-end);
- **Dense action annotations**—one row per atomic action, time-aligned to the video (*used in generation*);
- **Participant and session identifiers** (`participant_id`, `video_id`) so longitudinal behavior can be grouped (*used in generation*);
- **Timestamps** for when each action occurs (`narration_timestamp`, plus frame- and clock-based start/stop fields in the official CSV; *used in generation*, with start/stop fields optional);
- **Structured labels** (`verb`, `noun`, and their taxonomy classes) and a free-text **narration** describing the action in context (*used in generation*).

The Embodied Memory generator consumes only the annotation table (not pixels): it needs a time-ordered log of *who* did *what*, *when*, and *in which video*.

The release is restricted to an **EPIC-KITCHENS-100 extension video whitelist** (45 hours, 201 videos across 18 participants), so every item remains within the same longitudinal setting.

B.2 Question Types and Grounding

Design. Each item is *participant-specific*: it concerns one person's kitchen behavior across longitudinal sessions in the same home, not generic cooking knowledge, so the benchmark tests *consolidated user models* rather than generic cooking priors. Released questions require *evidence-grounded* stems and keys: every item must cite observable actions (timestamps, `video_ids`, verb–noun labels) drawn from EPIC narrations, so items remain auditable and cannot be answered from world knowledge alone. Because LLM generation is noisy and many candidates fail rule checks, we *overgenerate then curate* (target $\times 3$ per type before filtering) to hit per-type quotas without hand-writing questions; the surplus absorbs parse failures, weak evidence, and judge rejections. The pipeline applies a

cross-model judge (generator `gpt-5.5` vs. verifier `gpt-4o-mini`) so acceptance is not decided by the same model that proposed the item. Every released item must pass mandatory verifiability prompts at generation time plus post-hoc quality filtering (semantic deduplication, per-item verification, and answer re-derivation).

Scope of this appendix. This appendix specifies the annotation-grounded pipeline that produces the four reported EAM-QA types in Table 6. For each participant, we aggregate in-scope narrations, summarize them into compact statistics and per-video digests, prompt a generator LLM for structured multiple-choice items, and filter with a separate verifier model. Two earlier pixel-grounded types (object localization ELOC, temporal ordering EORDER) are excluded from the released benchmark because parametric and perception-heavy baselines already saturate those probes.

Construction steps. For each participant, the pipeline applies the following steps; each step is selected to keep generation *grounded* while fitting in context windows.

(i) Index. We aggregate verb/noun marginals, verb–object pairs, consecutive verb bigrams, and trigger→next histograms (within-video only). Marginals surface stable tool/object preferences for SPREF; trigger–response tables directly support SHABIT ("after X, what next?"); verb bigrams expose coarse workflow regularities for SROUTINE without sending the full action log to the LLM. These statistics are a *lossy but faithful* summary: they highlight recurring structure the generator should mine, while staying far smaller than raw narrations.

(ii) Summarize. Each in-scope video is compressed into a short Goal / key-sequence / object digest so cross-video types see thematic context (e.g., "checks pan by stirring") without loading thousands of raw rows.

(iii) Single-hop generate. Per-type prompts combine statistics, a sparse action sample, video digests, and shared instructions requiring timestamp and `video_id` citations, balanced distractors, and guards against tautological or commonsense-only items. SPREF, SHABIT, and SROUTINE use a lower decoding temperature than multi-hop variants to limit hallucinated counts while preserving phrasing diversity.

(iv) Multi-hop variants (within each type). Multi-hop is *not* a fifth question family: every headline type also receives 2-hop and 3-hop items. The pipeline first extracts 4–6 step action chains within a video, then prompts the generator to select a 2- or 3-step subchain and phrase a question whose answer depends on that subchain only. Deterministic chain extraction guarantees the hops exist in the log; the model selects and words the probe.

(v) Post-process. Rule deduplication removes exact duplicates; keyword topic deduplication is disabled in the reported release because it was too aggressive ($\sim$22% loss). LLM curation and contradiction resolution on SPREF/SHABIT drop near-duplicate or conflicting claims. A per-participant cap (50% per type) keeps the release balanced across cognitive classes. Long runs can resume from intermediate save points after interruption.

(vi) Construct E-correct controls. After fixing the 2,212 answerable items, we create 551 unanswerable controls (approximately 20% of the final panel) by deterministically sampling a question from a different donor participant, rewriting participant mentions to the target participant, and assigning option E as the correct answer. The answer agent receives only the target participant's memory bank, so none of the four donor-derived content options is supported there. The release records `is_unanswerable`, `source_participant`, and `target_participant` for auditing this construction.

Per-type inputs, flow, and prompts. Figure 10 and Table 6 summarize the pipeline and per-type inputs and evidence thresholds for each headline type. Every generation prompt appends the same three instruction blocks: **verifiability** (mandatory timestamps, `video_ids`, verb–noun citations), **answer format** (balanced distractors, varied correct-letter positions), and **anti-failure** guards (tautology, triviality, hallucination, stem leakage). Figure 11 gives the auditable prompt skeletons.

B.3 Quality Control and Diagnostics

Summarization and multi-hop wording. The summarization prompt requires each video digest to list a one-sentence goal, 2–5 titled arrow sequences, and key objects/tools. For multi-hop items in SPREF, SHABIT, SROUTINE, and ERECALL, the generator receives a timestamped within-video chain and must select a 2- or 3-step subchain before writing the question; released items are tagged with `reasoning_depth`$\in\{2,3\}$ but keep the parent type's schema.

Table 6: Headline question types, inputs, and evidence thresholds in the annotation-grounded Embodied Memory release. The dashed rule separates **semantic** probes consolidated across videos from ERECALL, which is **episodic** (video-specific). The generator's JSON schema may tag SHABIT/SROUTINE with `cognitive_level=procedural_memory` because habits and routines describe *how* someone acts; in our benchmark taxonomy they are nevertheless grouped with SPREF as declarative, cross-session semantic knowledge about the participant. Each type also has **multi-hop** variants (`reasoning_depth` $\in \{2, 3\}$): the same cognitive probe, but the answer requires chaining 2–3 within-video actions (not a separate question family).

Type	Class	What is tested	Primary inputs	Evidence required
SPREF	Semantic	Cross-video tool/method/location preference	Verb/object statistics; cross-video digests; sparse action sample	Preference supported in $\geq$2 videos with repeated observations
SHABIT	Semantic	After trigger X, typical next action (contextual habit)	Trigger$\rightarrow$next patterns; recurring action sequences; digests	Trigger in $\geq$2 videos; habit repeated often enough to be habitual
SROUTINE	Semantic	Multi-step workflow or kitchen strategy (routine)	Recurring multi-step workflows; verb-transition regularities; digests	Strategy seen in $\geq$2 videos; distractors are distinct approaches
ERECALL	Episodic	Main activity in a *specific* video	Per-video narrations $\rightarrow$ short activity label; plausible wrong activities	One episodic item per video; answer tied to that `video_id` only

Automated verification before release. Every candidate passes three filter stages, each targeting a different failure mode: **(1) Semantic dedup** (embedder in Table 7; cosine ≥ 0.95) groups near-duplicate stems before expensive judging; an LLM picks which cluster members to keep so we remove redundancy without collapsing legitimately distinct probes. **(2) Quality filter** (T=0): a type-aware YES/NO prompt rejects unclear stems, weak distractors, or answers not supported by cited evidence; a soft retry at T=0.3 rescues borderline items if the first pass is overly strict (<30% pass rate). **(3) Answer re-derivation**: the judge sees the question, choices, and evidence but *not* the stored label and must independently recover the same letter—catching cases where the generator assigned a key inconsistent with its own evidence. Rule-based gates at generation time enforce minimum support (e.g., $\geq$ 2 videos for cross-video types, minimum trigger counts for SHABIT) so obviously weak candidates never reach the judge. On a reference run for P06, 173 candidates survived in-generator steps and 129 were released after filtering (27 removed by the quality judge alone). Representative numeric thresholds are listed in Table 7.

OpenAI API configuration. The reported annotation-grounded release is produced through the **OpenAI Chat Completions API** (`https://api.openai.com/v1`). Each request is a single-turn user message (no system role, no dialogue history). We use `gpt-5.5` for generation and `gpt-4o-mini` for verification so acceptance is not self-judged. Structured outputs use JSON mode when the prompt requests JSON; the verifier uses low-temperature decoding (T=0, `top_p`= 0.9). The generator is a reasoning-style model: the API ignores custom temperature and instead allocates a larger completion budget so hidden reasoning tokens do not crowd out the visible answer. Operational settings are 3 concurrent requests, 240 s timeout, up to 3 retries with exponential backoff, and a 128k-token context budget with prompt truncation when needed.

Embedding dedup uses the embedder in Table 7 before any verifier API calls.

Reproducibility. We generate one bundle per participant on the extension video whitelist; per-type targets are 8 single-hop and 8 multi-hop items per semantic type (SPREF, SHABIT, SROUTINE, before $\times$3 overproduction) and one ERECALL item per in-scope video. Full launch parameters and record schemas are recorded in the MEMORA-Bench reproduction appendix.

B.4 Evaluation Protocol

Task and metrics. For each question q, the agent must choose one of five options $\{\mathrm{A}, \ldots, \mathrm{E}\}$. Options A–D are contentful answers; **E** denotes "information not available," so a model must recognize missing memory rather than guess among four plausible kitchen actions—otherwise parametric priors inflate accuracy on underspecified probes. **Accuracy** is reported per type and macro-aggregated over participants; the Pool column pools all headline items with count weighting. Each question is routed to the fixed memory bank of its declared tar-

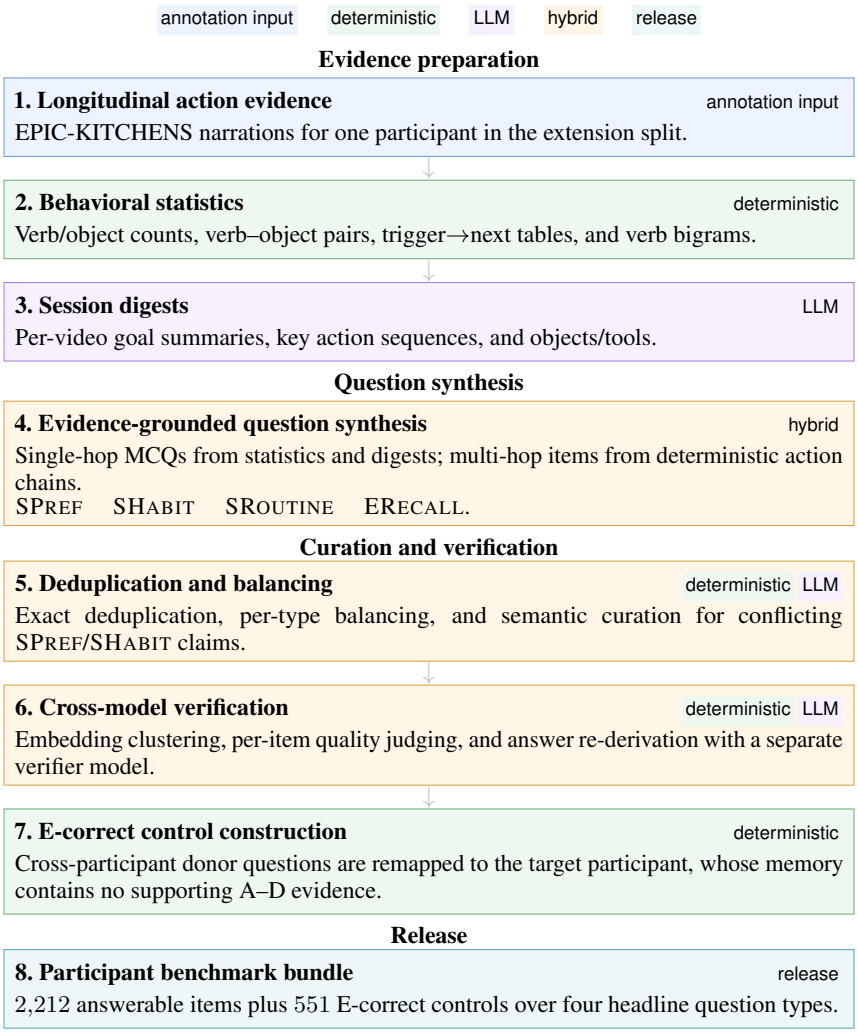


Figure 10: **MEMORA-Embodied Memory benchmark construction pipeline.** MEMORA-Bench converts longitudinal action annotations into participant-specific Embodied Memory questions through four phases: evidence preparation, question synthesis, curation and verification, and release. Colors indicate whether each stage is annotation input, deterministic processing, LLM-based synthesis, hybrid checking, or the released benchmark bundle.

get participant. Answerable items are grounded in that participant's observed corpus; E-correct controls are donor-derived questions for which the target memory intentionally contains no supporting content option. The released headline records do not define a per-question `ask_turn_id`, so this evaluation makes no claim of within-video prefix rollback.

Controlled comparison. All conditions share the same perception front-end, dense retrieval backend, answer-agent backbone, and ReAct budget within a column (Appendix F), so differences are not confounded by vision, embedding capacity, or inference budget. The compared memory interfaces are Parametric (no external memory), **Flat-1D** (Zeng et al., 2022) (chronological text memory exposed via a flat `search` tool), **Graph-2D** (an entity-relation graph memory instantiated in the style of M3-Agent (Long et al., 2025)), and **MEMORA** (typed substrates with consolidation variants). Cross-family gaps therefore reflect memory organization together with its intended read interface; raw/processed and episodic/full pairs provide the controlled tests of editing and consolidation.

Adapted memory-organization check. We additionally compare two recent organizational designs inside a matched MEMORA-Bench harness. Graph-2D is the entity-centric graph instantiation in the style of M3-Agent; the adapted MEMENTO organization (Kwon et al., 2026) partitions memory by participant and groups evidence around object semantics and recurring user patterns. Both adapta-

Shared verifiability rules

```
VERIFIABILITY REQUIREMENTS
In reasoning, cite:
1. timestamp(s): HH:MM:SS[.ms]
2. video_id(s) where evidence appears
3. exact verb-noun labels from input
4. counts for "typically" claims
Never fabricate unseen actions or times.
```

SPREF: preference

```
Goal: discover PERSONAL PREFERENCES.
Input: action statistics, sample actions, per-video
digests.
Require: preference in >=2 videos.
Discover: tool/order/location/method/timing.
Output fields: question, preference_type, activity_topic,
evidence.count, evidence.video_ids, choices,
correct_answer, reasoning.
```

SHABIT: contextual habit

```
Goal: find trigger -> response habits.
Input: statistics, common action sequences, video digests.
Require: trigger in >=2 videos; prefer trigger_count>=4.
Question form: "After [trigger], what does P typically do
next?"
Output: trigger_action, habit_count,
total_trigger_occurrences, video_ids, choices, answer.
```

SROUTINE: workflow strategy

```
Goal: identify strategic approaches, not generic step lists.
Input: recurring sequences, statistics, video digests.
Reject: universal flows such as pick-up->put-down or
open->close.
Require: seen_in_videos >= 2.
Choices: four fundamentally different strategies.
Output: task_topic, correct_sequence, evidence, choices,
answer.
```

ERECALL: video-specific episodic recall

```
Step A: summarize each video from narrations into short memory_summary + main_activity.
Step B: generate three plausible but wrong activities from varied categories.
Assemble one 4-way MCQ per video_id. Multi-hop ERecall uses the same schema but targets an outcome chain of depth 2 or 3
inside a single video.
```

Figure 11: **Prompt skeletons for Embodied Memory question generation.** All generation prompts append the shared verifiability rules; SPREF, SHABIT, SROUTINE, and ERECALL differ in the evidence pattern they require and the schema fields they emit. Placeholders `{P}` and `{N}` are filled per run, and multi-hop variants reuse the same framing with supplied action chains.

tions reuse the same upstream perception records, retrieval backend, answer backbone, and evaluation budget; the changed component is the organization presented to retrieval. This is an in-domain architectural adaptation, not an evaluation on the external MEMENTO benchmark, whose input consists of already processed memory entries. Table 8 reports the matched P01–P04 EAM-QA slice and the 153-task Generalize planning set.

Agent loop. The answering agent follows the shared Embodied Memory ReAct budget in Appendix F (up to five `search` calls, then one letter). For Embodied Memory evaluation, the harness loads the complete fixed memory bank for the question's target participant. No per-question temporal slicing is applied to the released headline panel; protection against unsupported answers is instead measured by the E-correct controls described above. Tool schemas and per-condition system prompts are fixed within each block; Graph-2D receives schema-equivalent per-type prompts so routing verbosity does not confound substrate comparisons.

Reporting diagnostics. A **Parametric Memory** column is always reported alongside structured memory: types where Parametric already exceeds $\sim$70% are flagged as prior-saturated in footnotes rather than excluded post hoc. E-selection rate is tracked separately because flat retrieval can overproduce information-not-available answers on some commercial backbones.

Illustrative items. Figure 12 shows the construction-time audit record corresponding to one released answerable item per headline type from participant P06. Evidence counts and reasoning depth are used during construction and quality filtering but are stripped from the public task record. The answering agent receives the question and the five released choices A–E, without the evidence or reasoning metadata.

B.5 Full and Conditional EAM-QA Evaluation Views

The Embodied Memory benchmark mixes *answerable* questions (ground truth $\in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$) with *unanswerable* probes (ground truth E), the latter required so a model cannot inflate accuracy by guessing on underspecified items. We report three views with explicit predicates. The **full benchmark** contains all 2,763 items. The **MEMORA-committed** view retains items for which MEMORA selects A–D. The **parametric-missed + MEMORA-committed** view additionally requires the Parametric condition to be incorrect. All compared conditions are scored on the same retained IDs within each backbone.

Table 7: Models and decoding settings for Embodied Memory benchmark generation. For `gpt-5.5`, listed T values document intent; the API uses its built-in sampling and an expanded completion cap.

Stage / role	Model	T	Max out.	Notes
Generator	`gpt-5.5`	—	—	Question synthesis and post-gen. curation
Verifier	`gpt-4o-mini`	0.0	varies	Per-item quality check and answer re-derivation
Embed dedup	`e5-base-v2` (Wang et al., 2024)	—	—	Local embedding model (not an API call)
Video summarization	`gpt-5.5`	0.1	512	Generation stage
SPREF/SHABIT/SROUTINE (single-hop)	`gpt-5.5`	0.3	6144	Generation stage
ERECALL activity label	`gpt-5.5`	0.4	400	Generation stage
ERECALL distractors	`gpt-5.5`	0.7	2048	Generation stage
Multi-hop (all four types)	`gpt-5.5`	0.7	6144	Generation stage
Post-generation curation	`gpt-5.5`	0.1	1000	Generation stage
Verifier: semantic-dedup pick	`gpt-4o-mini`	0.0	32	Verification stage
Verifier: quality filter	`gpt-4o-mini`	0.0	16	Verification stage
Verifier: answer re-derivation	`gpt-4o-mini`	0.0	32	Verification stage

Table 8: Matched adapted memory organizations. EAM-QA uses P01–P04 and the four released EAM-QA types; E-rate is option-E selection. Planning reports RGP on 153 Generalize tasks. Graph-2D is the M3-Agent-style organization used throughout our figures and tables, not an additional baseline.

Organization	EAM-QA Acc.	E-rate	RGP
Flat-1D	37.2	65.8	.386
Graph-2D	34.9	70.6	.336
MEMENTO (adapted)	39.8	48.3	.370
MEMORA	**42.0**	**35.7**	**.450**

These latter two views are conditional diagnostics, not alternative benchmark definitions. Parametric failure is a proxy for weak closed-book support, not a proof that a question is experience-dependent; likewise, selecting A–D is a commitment event, not by itself proof that the answer is grounded. The filter does not condition on MEMORA correctness, so an E-correct control on which MEMORA commits incorrectly remains in the diagnostic and is scored as an error. This makes the selection rule auditable while avoiding a correctness- conditioned subset. Table 9 reports all three views; the Parametric row is 0% in the final conditional view by construction and is therefore omitted from the per-backbone table.

Commit rate and commit-conditioned accuracy. Within the parametric-missed pool, we report the MEMORA commit rate $\mathrm{P}(\text{MEMORA picks A–D} \mid \text{parametric wrong})$ and commit-conditioned accuracy $\mathrm{P}(\text{correct} \mid \text{A–D, parametric wrong})$. The latter is the final conditional-view accuracy by construction. It rises from Gemma-4-26B-A4B 54.1% to Qwen3.6-27B 69.1% and Gemma-4-31B 74.5%, whereas commit rate varies non-monotonically (respectively 67.5%, 70.3%, and 50.7%). We therefore report both quantities rather than interpreting A–D selection alone as successful grounding. We read this as a *capacity-bound* rather than *pipeline-bound* cross-backbone pattern: absolute differences across backbones reflect the answer-agent's discriminative capacity once it has enough support to answer, not a failure of the memory pipeline that feeds it.

B.6 Per-backbone × per-type Results

The main paper headlines a 2×2 open-source cross-backbone design (Qwen 3.6 / Gemma 4 $\times$ MoE / dense; Table 3 displays the Gemma-4-31B-it column on the full 18-PID panel). This subsection unpacks the remaining three backbones — Qwen3.6-35B-A3B, Qwen3.6-27B, and Gemma-4-26B-A4B-it — under the unified parametric-missed, MEMORA-committed diagnostic of Appendix B.5 (Table 10; the per-backbone Δ_{consol} rows summarize the per-type lift).

Cross-backbone consistency of the consolidation lift. Under the unified parametric-missed, MEMORA-committed EAM-QA diagnostic (Table 10, Δ_{consol} rows), the consolidation lift Δ_{consol} = MEMORA − MEMORA-Episodic is positive on every question type and every backbone. The SROUTINE lift is the most robust as expected for the aggregation-requiring type: $+4.9$ on Qwen3.6-35B-A3B, $+9.9$ on Gemma-4-

SPREF: cross-video preference


```
{
  "qa_type": "user_preference",
  "participant_id": "P06",
  "question": "Across videos, when P06 checks food cooking in
      a pan, what observable method do they typically use?",
  "choices": [
    "A) Stir the pan contents",
    "B) Shake the pan briefly",
    "C) Leave pan untouched",
    "D) Pour food into bowl"
  ],
  "correct_answer": "A",
  "evidence": {
    "correct_option_count": 26,
    "video_ids": ["P06_101", "P06_102", "P06_103", "P06_104",
      "P06_105"]
  },
  "reasoning_depth": 1
}
```


SHABIT: trigger→response habit


```
{
  "qa_type": "contextual_habit",
  "participant_id": "P06",
  "question": "Across pan-checking moments, what does P06
      typically do immediately after picking up the spoon?",
  "choices": [
    "A) open drawer for another utensil",
    "B) stir pan before touching the lid",
    "C) take-off lid from the pan",
    "D) put-down spoon beside the hob"
  ],
  "correct_answer": "C",
  "evidence": {
    "correct_habit_count": 5,
    "total_trigger_occurrences": 15,
    "video_ids": ["P06_103", "P06_102", "P06_101"]
  },
  "reasoning_depth": 1
}
```


SROUTINE: cross-video workflow


```
{
  "qa_type": "action_pattern",
  "participant_id": "P06",
  "question": "During active hob cooking, what is P06's
      typical control routine?",
  "choices": [
    "A) Uses a spoon-led lid check, stirs, re-covers, then
      tastes.",
    "B) Shakes the pan with the lid on and avoids spoon
      testing.",
    "C) Transfers the contents to a bowl before checking
      doneness.",
    "D) Leaves the pan uncovered and adjusts heat without
      sampling."
  ],
  "correct_answer": "A",
  "evidence": {
    "seen_in_videos": ["P06_103", "P06_102", "P06_101"],
    "occurrences": 3
  },
  "reasoning_depth": 1
}
```


ERECALL: video-specific episodic recall (d=2)


```
{
  "qa_type": "event_recall",
  "participant_id": "P06",
  "video_id": "P06_104",
  "question": "In video P06_104, P06 handled the sweet potato.
      What did they do with it?",
  "choices": [
    "A) They mashed the sweet potato and put it into the pan
      .",
    "B) They cut the sweet potato and left it beside the pan
      .",
    "C) They cut the sweet potato and put it in the pan.",
    "D) They pierced the sweet potato and kept it on the fork
      ."
  ],
  "correct_answer": "C",
  "evidence": {
    "video_id": "P06_104",
    "core_chain": [
      {"action": "cut potato:sweet", "idx": 5},
      {"action": "put-in potato:sweet", "idx": 6}
    ]
  },
  "reasoning_depth": 2
}
```


Figure 12: Construction-time audit views of released Embodied Memory items (P06). Left column: SPREF and SROUTINE; right column: SHABIT and ERECALL. For compact display the figure shows only content options A–D; every released JSON record already contains option E (information not available), while the displayed evidence metadata is omitted from the public task input.

26B-A4B, +10.7 on Qwen3.6-27B, and +24.6 points on Gemma-4-31B. SHABIT, the other cross-event aggregation type, lifts by +6.1 to +13.7 points across the four backbones; the single-fact retrieval types lift by +7.0 to +11.3 points (SPREF) and +4.7 to +10.5 points (ERECALL). The four-backbone means are +12.5/+10.1/+9.2/+7.4 points on SROUTINE/SHABIT/SPREF/ERECALL, with every per-backbone, per-type cell positive, consistent with consolidation strengthening retrieval whenever the question can be committed to a contentful answer.

Scaling to larger corpora. The pipeline in Figure 10 is *participant-agnostic*: adding more benchmark volume means repeating steps 1–7 for additional people or sessions, not redesigning the question types. To extend beyond the current extension split, a new corpus must supply at minimum:

- **Participant identifiers** that are stable across sessions (so cross-video SPREF, SHABIT, and SROUTINE items remain meaningful);
- **Video/session identifiers** mapping each file to exactly one participant;
- **Time-ordered action annotations** per video: timestamps plus verb–object (or equivalent) labels, and preferably short natural-language descriptions for audit trails;
- A defined **video whitelist** (or split policy) stating which sessions enter generation;

Table 9: **EAM-QA evaluation views: MEMORA overall accuracy (%) and gap to strongest non-MEMORA condition** (FLAT-1D/GRAPH-2D, raw or after MEMORA's online editing pass; MEMORA-Episodic excluded as it is a MEMORA ablation rather than a baseline). The full-benchmark view reports all EAM-QA questions; the MEMORA-committed view retains cases where MEMORA selects A–D; the final conditional view additionally requires the parametric baseline to be incorrect.

	Q3.6-35B-A3B$^{\sharp}$		**G4-26B-A4B**		**Q3.6-27B**		**G4-31B**	
Evaluation view	MEMORA	Δbest	MEMORA	Δbest	MEMORA	Δbest	MEMORA	Δbest
Full benchmark	50.2	-3.4	39.3	-9.9	51.7	$+1.5$	46.1	-0.5
MEMORA-committed	56.5	-0.3	45.8	-7.2	61.3	$+5.9$	67.7	$+9.6$
Parametric-missed + MEMORA-committed	**63.7**	$\mathbf{+11.0}$	**54.1**	$\mathbf{+9.6}$	**69.1**	$\mathbf{+19.3}$	**74.5**	$\mathbf{+20.5}$
Net swing (full $\rightarrow$ headline)	$+14.4$ points		$+19.5$ points		$+17.8$ points		$+21.0$ points	

Subset sizes per backbone ($n_{\text{full}}/n_{\text{answerable}}/n_{\text{headline}}$): Q3.6-35B-A3B $1{,}202/819/721$; G4-26B-A4B $2{,}763/1{,}770/1{,}497$; Q3.6-27B $2{,}763/1{,}753/1{,}556$; G4-31B $2{,}763/1{,}234/1{,}122$. Δbest = MEMORA $-$ best of {FLAT-1D raw/processed, GRAPH-2D raw/processed} on the same subset. $^{\sharp}$ Q3.6-35B-A3B panel is reported on the 8-PID intersection (n_{full}=1,202) for which this diagnostic was first computed; the headline row agrees with the 12-PID Qwen3.6-35B-A3B intersection in Table 10 (MEMORA 63.5%, $+11.8$ points over the strongest non-MEMORA condition) within ± 0.8 points.

- For **evaluation** on new footage: synchronized RGB video and the same perception/retrieval stack as in the main experiments (generation can remain annotation-only).

Optional enrichments—frame-level bounding boxes, object tracks, or a parallel VLM-augmented track—are not required for the headline annotation-grounded release but would enable spatial types (ELOC) or pixel-grounded verification at larger scale. As annotation density and participant count grow, the same overgenerate-and-filter recipe applies; cost scales with LLM verifier calls per candidate, which is why semantic dedup and rule-based gates precede the cross-model judge.

Table 10: **MEMORA-Embodied Memory Assessment accuracy (%): per-backbone, per-type panel**. All four backbones report under the parametric-missed, MEMORA-committed EAM-QA diagnostic (Appendix B.5): **Gemma-4-31B-it** on the full 18-PID panel (N=1,122; mirrors Table 3); **Qwen3.6-35B-A3B** on the 12-PID intersection where all seven conditions are currently materialized (N=1,141); **Qwen3.6-27B** on the full 18-PID panel (N=1,556); **Gemma-4-26B-A4B-it** on the full 18-PID panel (N=1,497). Δ_{consol} row: MEMORA minus MEMORA-Episodic. The Parametric (no-memory) row is omitted because the diagnostic retains only questions it answers incorrectly (0% by construction); full-benchmark and MEMORA-committed results are reported in Table 9.

		Cross-video (semantic)			Single-video	
Backbone	**Memory condition**	**SPREF**	**SHABIT**	**SROUTINE**	**ERECALL**	**Overall**
Qwen3.6-35B-A3B[♭] *(MoE, 64K ctx)* *(N=1,141)*	Flat-1D (raw)	51.2	33.5	53.2	12.5	41.6
	Flat-1D (processed)	53.1	34.2	52.9	5.3	41.3
	Graph-2D (raw)	50.6	27.4	54.7	48.7	45.0
	Graph-2D (processed)	61.5	34.2	59.6	50.7	51.7
	MEMORA-Episodic	64.3	42.2	63.5	53.3	56.3
	MEMORA	**73.3**	**48.3**	**68.4**	**63.8**	**63.5**
	Δ_{consol}	**+9.0**	**+6.1**	**+4.9**	**+10.5**	**+7.2**
Qwen3.6-27B[♭] *(dense, 64K ctx)* *(N=1,556)*	Flat-1D (raw)	48.9	41.6	49.6	65.7	49.6
	Flat-1D (processed)	49.8	38.7	49.4	61.0	48.5
	Graph-2D (raw)	51.1	29.6	53.3	53.5	46.8
	Graph-2D (processed)	55.1	32.7	55.4	55.9	49.7
	MEMORA-Episodic	64.1	48.8	64.0	68.5	60.9
	MEMORA	**71.1**	**57.1**	**74.8**	**73.2**	**69.1**
	Δ_{consol}	**+7.0**	**+8.3**	**+10.7**	+4.7	**+8.2**
Gemma-4-26B-A4B[♭] *(MoE, 64K ctx)* *(N=1,497)*	Flat-1D (raw)	51.0	28.0	42.0	59.4	42.4
	Flat-1D (processed)	53.9	31.3	42.2	62.4	44.5
	Graph-2D (raw)	40.4	16.3	31.6	60.2	32.5
	Graph-2D (processed)	48.8	23.4	40.8	60.2	40.1
	MEMORA-Episodic	45.3	39.2	39.6	62.4	43.2
	MEMORA	**56.5**	**51.7**	**49.5**	**70.7**	**54.1**
	Δ_{consol}	**+11.3**	**+12.4**	**+9.9**	**+8.3**	**+10.9**
Gemma-4-31B[♭] *(dense, 64K ctx)* *(N=1,122)*	Flat-1D (raw)	63.2	36.8	53.8	51.2	52.0
	Flat-1D (processed)	62.4	41.4	56.8	53.0	54.0
	Graph-2D (raw)	50.9	21.4	37.1	55.4	40.0
	Graph-2D (processed)	57.1	27.0	48.6	57.1	47.0
	MEMORA-Episodic	67.4	47.0	56.8	73.8	60.1
	MEMORA	**76.8**	**60.7**	**81.5**	**79.8**	**74.5**
	Δ_{consol}	**+9.4**	**+13.7**	**+24.6**	**+6.0**	**+14.4**

Bold: best per column within each backbone block. Δ_{consol} values are bolded when they exceed the ±4–5 points per-type binomial 95% CI. [♭] Parametric-missed, MEMORA-committed diagnostic (Appendix B.5): Gemma-4-31B-it on the full 18-PID EAM-QA panel (N=1,122; mirrors Table 3); Qwen3.6-35B-A3B on the 12-PID intersection (N=1,141); Qwen3.6-27B on the full 18-PID panel (N=1,556); Gemma-4-26B-A4B-it on the full 18-PID panel (N=1,497). The Qwen3.6-35B-A3B row is reported on the 12-PID intersection because that is the largest subset in the public release on which all seven memory conditions are materialized. The 8-PID sensitivity subset reported in the [♯] footnote of Table 9 confirms that MEMORA's accuracy and its +11.8-point lead over the strongest controlled baseline on Qwen3.6-35B-A3B are preserved within ±0.8 points across the 8-PID and 12-PID subsets.

C MEMORA-Planning: Construction and Evaluation Protocol

Appendix C specifies the prospective planning evaluation in MEMORA-Bench. **MEMORA-Planning** evaluates whether the constructed memory bank supports future action, rather than only retrospective question answering. In contrast to the Embodied Memory assessment, which asks an agent to select a discrete answer about past experience, MEMORA-Planning requires the agent to synthesize an ordered kitchen plan for a specific participant. Each item is grounded in EPIC-KITCHENS-100 annotations (Damen et al., 2022): the reference sequence is an observed segment of narrated actions, but the planner is evaluated on whether it can produce an executable, participant-specific procedure rather than reproduce the annotation verbatim.

The reported benchmark uses the released MEMORA-Planning split. Its tasks are generated deterministically from EPIC action annotations, restricted to videos available in the participant's consolidated **Inferred Knowledge**, and enriched with object metadata from the same memory. No LLM is used to generate task labels or task queries in this release. The evaluation compares no-memory, linear-memory, and structured-memory agents under the same planner model, and reports rule-based metrics to avoid judging an LLM agent with another LLM.

C.1 Task Construction

Task definition. An instance consists of a natural-language request x, a participant ID, a video ID, and a reference sequence of EPIC actions. The request has the form of a user instruction to a household robot, e.g., "Help P03 clean the washing up bowl in the sink." The reference sequence stores the ordered EPIC narrations, verbs, nouns, and timestamps for the segment from which the request was derived. The benchmark therefore tests procedural generation under verifiable evidence: every released item can be audited against the annotation table, but the model's output remains open-ended.

Design principles. Planning items are *participant-specific*: they should reward agents that retrieve the person's observed routines, object locations, and preferences, not agents that merely know generic cooking scripts. The generation process is *model-agnostic*: segmentation, classification, query templating, diversity selection, filtering, and memory grounding are rule-based in the reported release. Finally, the benchmark is *memory-scoped*: a task is generated only from videos present in that participant's consolidated **Inferred Knowledge**, so a memory-enabled agent is not penalized for lacking access to the relevant episode. Figure 13 summarizes the deterministic generation stages.

Segmentation and task typing. Within each video, action rows are sorted by timestamp. A hard split is placed when adjacent narrations are separated by more than 30 seconds. Soft splits are deterministic: verbs map to `cleanup`, `cooking`, `prep`, or `general`, and action i starts a new segment only if the $i-1 \to i$ gap exceeds 10 seconds, the category changes between two non-`general` categories, and action $i+1$ confirms the new category. Segments with fewer than three actions are merged with a neighbor, segments with more than 15 actions are divided into near-even chunks, and the resulting fixed task set is shared by all memory conditions. The candidates are then classified into the five task types in Table 11. Filler pick-and-place verbs are removed before classification so that superficial object motion does not dominate the task label.

Selection and memory-grounded enrichment. Candidate tasks are scored by proximity to the target length, verb diversity, and the fraction of non-filler actions. For each participant we greedily select up to 12 diverse tasks, with a per-type cap of four tasks under the main-paper configuration. A subsequent improvement pass computes duplicate-narration rate, $1 - |\text{unique narrations}|/|\text{steps}|$, and moves tasks above 0.70 to an audit-only filtered bucket. For retained tasks, each EPIC noun is matched against the same video's `object_registry` using exact, substring, token-overlap, and synonym matching. Successful matches add an enriched narration and a structured `matched_object` record containing memory object ID, name, color, material, and location. This enrichment is strictly additive: the original EPIC fields remain unchanged.

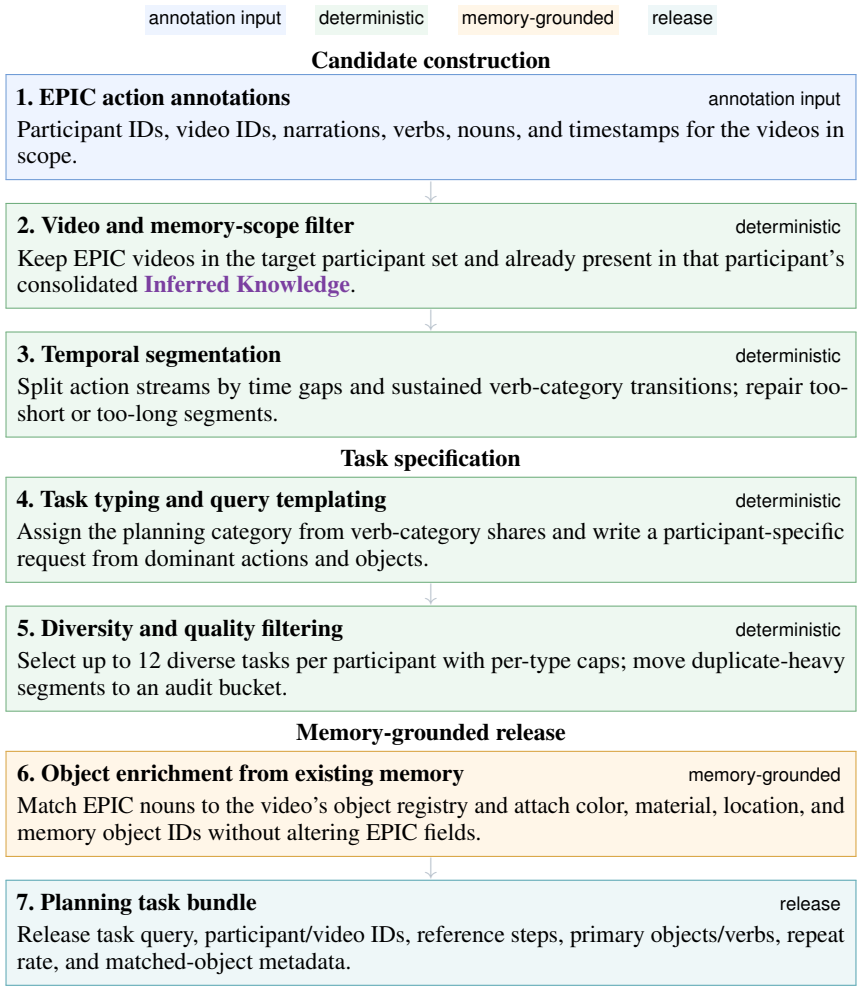


Figure 13: **MEMORA-Planning benchmark construction pipeline.** The flow converts EPIC action annotations into prospective, participant-specific planning requests. Unlike the memory-bank construction flow in Figure 6, this figure describes held-out benchmark construction; the memory bank is only read during video scope checking and object enrichment.

Table 12: Core schema for an improved planning task. The original EPIC fields remain available after memory enrichment.

Field	Source	Description
`task_id`	rule	`plan_{video_id}_{segment}`
`participant_id`, `video_id`	EPIC	Participant and session identifiers
`task_type`	rule	One of the five labels in Table 11
`task_query`	rule	Natural request produced from task type and dominant objects/actions
`ground_truth_steps`	EPIC	Ordered action rows with narration, verb, noun, start, and stop times
`primary_objects`, `primary_verbs`	rule	Top segment nouns and verbs
`repeat_rate`	rule	Duplicate-narration fraction used by the filter
`enriched_narration`	memory + rule	Optional step text augmented with object attributes
`matched_object`	memory + rule	Optional memory object ID, name, color, material, and location

Consolidated Inferred Knowledge. The MEMORA conditions consume a per-participant consolidated Inferred Knowledge bank. For planning, the most important fields are `skill_memory.routine_skills`, which stores routine goals, canonical steps, supporting episodes, and key objects; and `skill_memory.generated_preferences`, which stores evidence-grounded preferences, keywords, confidence scores, and supporting episodes. These fields make the consolidated Inferred Knowledge more than a searchable transcript: they expose consolidated procedures and preferences that a planner can retrieve and adapt.

Participant coverage. The reported generation covers 18 EPIC participants with released consolidated Inferred Knowledge banks: P01, P02, P03, P04, P06, P07, P09, P11, P12, P22, P23, P25, P26, P27, P28, P30, P35, and P37. P05 is excluded because its EPIC identifier convention does not match the three-digit suffix pipeline used for this release.

Table 11: MEMORA-Planning task taxonomy. Categories are assigned by deterministic verb-category thresholds.

Task type	Trigger	Example request
`meal_preparation`	preparation verbs $> 30\%$	Help P02 peel and prepare the onion.
`cleanup_organization`	cleanup verbs $> 35\%$	Help P03 clean the washing up bowl in the sink.
`object_retrieval_setup`	setup verbs $> 30\%$, or storage nouns with no meaningful verb	Help P06 get items from storage and set up.
`multi_step_cooking`	cooking verbs $> 30\%$	Help P12 stir-fry the vegetables.
`routine_reproduction`	no dominant meaningful category	Help P11 complete the kitchen routine.

C.2 Memory Conditions and Tools

Evaluation settings. The evaluation compares seven settings while holding the planning model, benchmark tasks, perception stream, agent budget, and shared runtime stack (Appendix F) fixed. Each representation is paired with its intended read tools and retrieval instructions. Cross-family results therefore compare complete memory interfaces; the raw/processed and episodic/full pairs provide the more tightly controlled tests of online editing and offline consolidation, respectively. The linear-memory baseline follows the Socratic-models style of exposing a text memory through language-model interaction (Zeng et al., 2022). Table 13 summarizes the conditions.

Table 13: Planning evaluation settings. Cross-family rows differ in memory representation and read interface; paired rows within a family isolate editing or consolidation.

Condition	Memory	Tools	Prompt behavior
Parametric	none	none	Single-shot planning from the request only.
Flat-1D (raw)	chronological observations	`search`	Search-first access to unedited text memory.
Flat-1D (processed)	chronological edited observations	`search`	Same Flat-1D interface after online memory editing.
Graph-2D (raw)	entity-relation graph	graph-oriented entity, activity, and temporal retrieval	Procedure retrieval followed by entity grounding.
Graph-2D (processed)	graph rebuilt from edited observations	same Graph-2D tools	Same Graph-2D interface after online memory editing.
MEMORA-Episodic	**Environment Memory**, **Entity Memory**, and **Activity Memory**	entity, activity, temporal, and unified retrieval	Episodic evidence followed by entity grounding.
MEMORA (full)	all four typed stores	episodic retrieval plus routine and preference retrieval	Routine-first retrieval followed by preference and entity grounding.

Planner and runtime. Planning-specific runtime settings beyond the shared stack in Appendix F are:

- **Task-list loading:** the planner is loaded once per 72-specification run so weights stay resident for the full batch.
- **ReAct budget:** at most 8 iterations, including a forced-answer final iteration with tools disabled.
- **Tool parsing:** a local parser handles Qwen-style XML/tool-call formats inside the agent wrapper.
- **Optional judge:** `gpt-4o-mini` is available for ablations but is not used in the reported rule-based panel.

ReAct loop and plan parser. Memory-backed conditions run a ReAct-style loop (Yao et al., 2023) under the shared agent-loop budget in Appendix F (Planning: 8 iterations including forced answer). At each iteration the environment supplies conversation state and registered tools; valid tool calls are executed and appended as observations. On the final iteration, the environment enters forced-answer mode: tools are cleared, previous tool calls are summarized as text, and the prompt explicitly requires a final `Plan:` rather than another tool call. The parser extracts the generated plan from numbered lines, bullet lists, or an explicit `Plan:` block.

Structured planning tools. The MEMORA prompt is designed to make the consolidated **Inferred Knowledge** operational. It asks the agent to begin with `get_routine_skill(goal_query, top_k)`, optionally query `get_preferences`, and then ground concrete objects through **Entity Memory** search. `get_routine_skill` retrieves routine memories by embedding similarity over routine goals, canonical steps, and key objects. `get_preferences` retrieves evidence-grounded preferences by query similarity or confidence.

Planning prompt templates. Figure 14 summarizes the operational differences among the prompt

Table 14: **Expanded memory-formation-to-planning examples.** This is the complete version of main-paper Table 2: each row preserves the input evidence, MEMORA record or retrieval sequence, resulting output, planning relevance, and controlled contrast.

Capability tested	Real MEMORA evidence/output	Why it matters
Maintain an object through time *Input evidence:* the same dirty plate is observed at 40 s, 50 s, and 540 s.	**Entity Memory record:** `plate_with_food_residue` **State history:** `idle@countertop` (40 s) → `idle@countertop` (50 s) → `being_used@countertop` (540 s). The editor keeps one object identity while recording state changes.	**Planning relevance:** future actions need the current state of a persistent object, not a pile of repeated sightings. **Controlled contrast:** append-only memory stores the three sightings separately. Online editing reduces entity records by median $\approx 18\times$ (Figure 2).
Turn remembered evidence into action *Goal:* Help P03 clean the washing up bowl in the sink.	**Typed retrieval** 1. `get_routine_skill("clean a bowl")` returns an under-specific cookware routine, which MEMORA rejects. 2. `get_routine_skill("wash bowl")` retrieves the dishwashing routine. 3. `search_objects(name="bowl")` grounds the black cooking bowl, green sponge, chrome faucet, and metal drying rack. **Grounded plan:** clean the black cooking bowl with the green sponge under the chrome faucet, then place it on the metal drying rack.	**Planning relevance:** the agent needs to bind a generic goal to experience-grounded objects, tools, and places. **Controlled contrast:** No Memory fills in generic objects; Flat-1D retrieves plausible but mismatched bowl/towel evidence.
Consolidate repeated experience *EAM-QA:* SPREF. When preparing to cook eggs, does P04 prefer a spatula or fork? Choices: A) random, B) fork, C) spatula, D) whisk, E) unavailable. Gold: C.	**Answer: C. Evidence:** P04 repeatedly stirs yellow mixtures/eggs with a wooden spatula (e.g., P04_116 turns 28, 39).	**Planning relevance:** plans can reflect stable participant preferences when repeated evidence supports them. **Controlled contrast:** No Memory, Flat-1D, and Graph-2D select E despite repeated action evidence.

families and the shared forced-answer mode.

No memory

```
System: robot planning assistant.
Input: task_query and participant_id.
Constraint: do not assume participant-specific memory.
Required final answer: ordered executable plan with
numbered steps.
```

Flat-1D memory

```
System: robot planning assistant with access to text memory.
Tool: search(query).
Instruction: retrieve relevant observations before planning
and ground the final plan in retrieved evidence when
available.
```

Graph-2D memory

```
System: robot planning assistant with access to an
entity-relation graph.
Tools: graph-oriented entity, activity, and temporal
retrieval.
Instruction: retrieve procedural evidence, then ground
named entities and locations.
```

MEMORA-Episodic

```
System: robot planning assistant with typed episodic memory.
Tools: entity, activity, temporal, and unified search.
Instruction: recover relevant episodes, then ground
entities and locations.
```

MEMORA

```
System: robot planning assistant for a specific person's
kitchen.
Priority: call get_routine_skill first; call
get_preferences when preferences may affect the plan; use
object tools for grounding; use generic search only as
fallback.
```

Forced final answer

```
Instruction: tool use is disabled. Do not emit tool-call
XML or analysis tags. Return only a parseable final plan.
Output: Plan: 1. ... 2. ...
```

Figure 14: **Planning prompt families.** The panel records the operational differences in memory, tools, and retrieval priority; every condition shares the same final-answer constraint and iteration budget.

C.3 Metrics

Rule-based metrics. The reported panel uses rule-based metrics rather than an LLM judge. For a generated plan $P = (p_1, \ldots, p_m)$, task query x, participant routines R, and participant preferences Q, we compute four scores.

Table 15: Rule-based planning metrics. Undefined routine- or preference-based scores are skipped when computing means.

Metric	Question	Evidence	Undefined if
Exec	Is each step physically actionable?	Generated plan text	Plan is empty or unparseable
OrderExec	Are reference steps recovered in order?	Generated plan + reference step sequence (Replay routine or Generalize GT)	Reference unavailable
KeyObj	Does the plan mention key routine objects?	Matched routine key-object list	No matched routine or no key objects
KBGround	Are plan objects part of the participant's KB vocabulary?	Generated plan + KB noun set	Plan contains no kitchen-noun candidate
PrefAdh	Does the plan reflect relevant preferences?	Matched preference entries	No preference passes the relevance threshold
RGP	Composite mean of OrderExec, KeyObj, and PrefAdh	Condition-level means of the defined axes above	All component axes are undefined

Reference provenance. Replay asks whether an agent recovers a workflow already present in participant memory, so OrderExec, KeyObj, and PrefAdh use matched consolidated routine, object, and preference references. Generalize has no matching stored routine by construction; procedure order is therefore scored against verified task-level `ground_truth_steps`, with the selected source recorded in `metric_source`.

Executability (Exec). A step is executable if it contains an action, an object, and a location phrase according to rule-based predicates. Let $A(p)$, $O(p)$, and $L(p)$ denote the action, object, and location predicates:

$$\mathrm{Exec}(P) = \frac{|\{p_i \in P : A(p_i) \wedge O(p_i) \wedge L(p_i)\}|}{|P|}. \tag{4}$$

Procedure order (OrderExec). The metric compares the normalized action-verb sequence of the generated plan with a per-task reference verb sequence. On Replay, the reference is the matched routine's canonical-step verbs (selected as the participant's `routine_skills` entry with highest Jaccard overlap with the task query, threshold 0.10). On Generalize, where no matching routine exists by construction, the reference is the task's verified `ground_truth_steps` verb sequence; the metric router (`scripts/compute_rule_metrics.py`) auto-detects Generalize tasks by their `denovo_` prefix and records the chosen reference in a per-task `metric_source` field.

$$\mathrm{OrderExec}(P, s) = \frac{\mathrm{LCS}(v(P), v(s))}{|v(s)|}, \tag{5}$$

where s is the per-task reference sequence and $v(\cdot)$ denotes a synonym- and inflection-normalized verb sequence. Tasks with no qualifying reference are undefined and skipped in the metric mean.

Key-object coverage (KeyObj). For the matched routine r with key-object set K_r, we compute:

$$\mathrm{KeyObj}(P, r) = \frac{|\{k \in K_r : k \in P\}|}{|K_r|}. \tag{6}$$

Preference adherence (PrefAdh). Preferences are first filtered by Jaccard overlap between the task query and the preference text or keywords, with threshold 0.05. For each relevant preference, we measure the fraction of preference tokens appearing in the plan, then average over relevant preferences.

Robot-Grounded Plan score (RGP). The headline planning aggregate is the unweighted mean of three condition-level robot-followability scores. For condition c, each component is first averaged over the tasks on which that component is defined:

$$\mathrm{RGP}(c) = \tfrac{1}{3}\big(\overline{\mathrm{OrderExec}}_c + \overline{\mathrm{KeyObj}}_c + \overline{\mathrm{PrefAdh}}_c\big). \tag{7}$$

Each axis captures one execution requirement (order, task-relevant object coverage, personalization). Raw Exec is reported separately because it rewards plan length regardless of correctness. KBGround is reported as a separate diagnostic in §C.6 rather than folded into RGP, for the metric reasons discussed there.

Verb normalization. OrderExec uses the V2 verb extractor for both the plan and reference sequences. This normalization corrects two systematic under-estimation errors: canonical memory steps often begin with "Person ..." rather than "The person ...", and routine or GT-step descriptions often use inflected verbs such as *places*, *washing*, or *preparing*. V2 strips article-free and pronominal subjects, maps common inflections to canonical verb classes, and suppresses non-verb fallback tokens such as *person*. The LCS scoring formula itself is unchanged.

C.4 Results and Diagnostics

Cross-backbone summary. Figure 15 summarizes the eight (backbone, benchmark) cells from

the main paper. Grey = best non-MEMORA baseline; orange = best MEMORA variant. MEMORA wins every cell, with Δ RGP rising from $+0.012 - +0.056$ on Replay to $+0.035 - +0.075$ on Generalize. Two recurring patterns underlie the two observations summarised in §5.3. First, MEMORA's gain is larger on Generalize than on Replay for three of the four backbones, suggesting that its benefits are often greatest when the model must compose beyond a matching observed episode. Second, the two larger reasoning backbones (Gemma-4-31B-it and Qwen3.6-35B-A3B) extract more value from the *same* memory state on Generalize, both delivering $\geq +0.064$ absolute Δ RGP, consistent with a reasoning-bound rather than memory-bound ceiling on the harder split.

Per-condition numerical results: Gemma-4-31B-it (main-paper RGP figure). Figure 3 reports per-condition RGP on Gemma-4-31B-it for Replay (n=207) and Generalize (n=153); the per-axis OrderExec / KeyObj / PrefAdh decomposition referenced in §5.3 is available in the figure companion panel.

Per-condition numerical results: Qwen3.6-35B-A3B. Table 16 reports the same per-condition decomposition on Qwen3.6-35B-A3B. The qualitative pattern matches Gemma-4-31B-it: typed retrieval lifts the same three axes, with the magnitude consistently larger on Generalize, where chunk-retrieval baselines no longer benefit from verbatim segment recall.

Per-store tool-use distribution. Figure 16 aggregates every logged `tool_call` across all 1,440 planning runs (4 backbones $\times$ (207 Replay $+153$ Generalize tasks), 7,163 calls under the main MEMORA configuration) and maps each tool to the typed store it queries (**Entity Memory**: `search_objects`, `get_object_history`; **Inferred Knowledge**: `get_routine_skill`, `get_preferences`; **Activity Memory**: `search_activities`). Two patterns recur. First, backbones differ sharply in how they spend their tool budget: **Inferred Knowledge** share ranges from 27% (Qwen3.6-27B) to 54% (Gemma4-26B-A4B-it), suggesting that a single-store interface may fit backbone families unevenly. Second, comparing MEMORA against the MEMORA-Episodic ablation (identical agent loop and KB, but agents rely on **Entity Memory** and **Activity Memory** tools only) shows that **Inferred Knowledge**'s 40% share does *not* simply vanish: 35% of all tool calls instead migrate to `search_activities` over **Activity Memory**, consistent with consolidated routines absorbing queries that otherwise shift to episodic activity search. Across both configurations the unified `search` dispatcher remains rare ($< 4\%$); agents prefer the typed entry points the prompt advertises, and $\approx 98.5\%$ of MEMORA's tool calls go through the typed cascade (`search_objects` / `get_routine_skill` / `get_preferences` / `search_activities`), with the cascade invoked on 96–100% of Generalize tasks per backbone. For contrast, Flat-1D conditions on the same Generalize split issue an average of 5.8 `search` queries per task, overwhelmingly keyword lookups that can surface a verbatim chunk when the task matches an observed video but offer no structured affordance for composition.

Illustrative trace. Figure 17 provides a compact excerpt from a real MEMORA trace in the reported evaluation. The task is `plan_P03_101_019`: "Help P03 clean the washing up bowl in the sink" (`cleanup_organization`, 4/8 iterations). The agent first retrieves a generic cookware-washing routine, notices that the retrieved key objects do not match the requested bowl, refines the routine query, grounds the bowl through object search, and then emits a grounded plan.

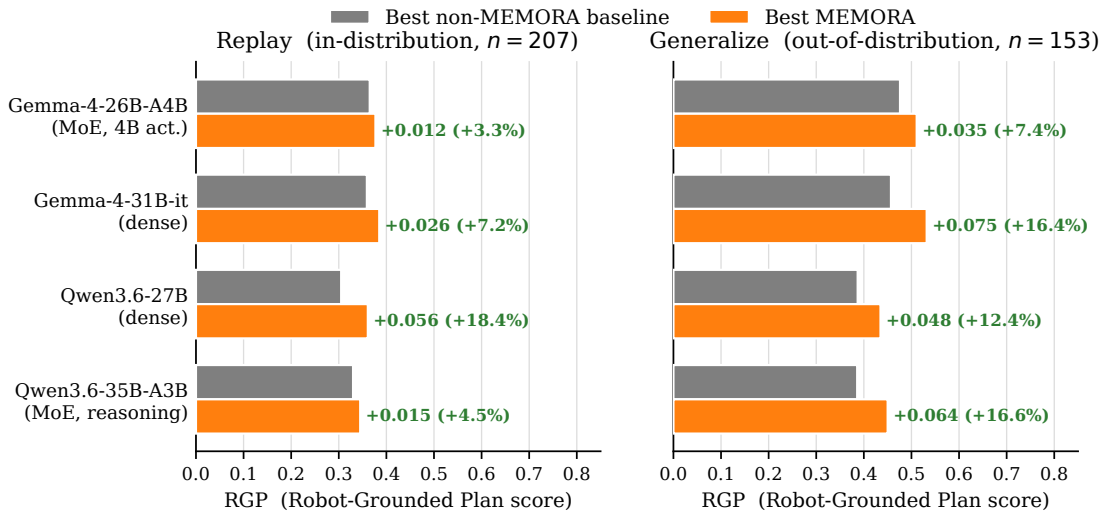


Figure 15: **Cross-backbone RGP summary** (companion to Figure 3). Grey = best non-MEMORA baseline; orange = best MEMORA variant; annotations give Δ RGP and relative gain.

Table 16: Per-condition per-axis Planning detail on Qwen3.6-35B-A3B. *(proc.)* indicates a baseline substrate rebuilt after MEMORA's online editing pass. **Bold**: best per column within each benchmark block.

	Replay (n=207)				**Generalize** (n=153)			
Condition	OrderExec	KeyObj	PrefAdh	RGP	OrderExec	KeyObj	PrefAdh	RGP
Parametric (No Memory)	.072	.316	.233	.207	.175	.485	.301	.320
Flat-1D (raw)	.087	.473	.382	.314	.166	.570	.422	.386
Graph-2D (raw)	.065	.411	.386	.287	.145	.491	.371	.336
Flat-1D (proc.)	**.096**	.494	.400	.330	.170	.531	.405	.369
Graph-2D (proc.)	.079	.426	.397	.300	.151	.492	.425	.356
MEMORA-Episodic	.085	**.501**	.448	**.345**	.186	**.651**	.440	.425
MEMORA (full)	.091	.474	**.450**	.338	**.243**	.650	**.458**	**.450**

```
Task: Help P03 clean the washing up bowl in the sink.
Iter. 1: get_routine_skill(goal_query="clean a bowl in
sink", top_k=3) returns a cookware-washing routine about
pot/spoon/stove, so the agent rejects it as under-specific.
Iter. 2: get_routine_skill(goal_query="wash a bowl",
top_k=2) retrieves a dishwashing routine mentioning a black
cooking bowl and drying rack.
Iter. 3: search_objects(name="bowl") returns bowl attributes
and nearby grounding objects including green sponge, chrome
faucet, and metal drying rack.
Final plan excerpt: pick up the black cooking bowl from the
sink; pick up the green sponge; turn on the chrome faucet;
clean the bowl; place it on the metal drying rack next to
the sink.
```

Figure 17: **Excerpt from a real `memora` ReAct trace for `plan_P03_101_019`.** The trace illustrates routine-first retrieval followed by object grounding before final plan emission.

C.5 Generation pipeline reproducibility

Reproducibility. The generation pipeline is deterministic under the fixed EPIC CSV, participant list, and released consolidated **Inferred Knowledge** banks. Task extraction targets 12 tasks per participant, followed by a rule-based improvement pass that computes duplicate-narration rate and moves high-repeat segments to an audit-only bucket. The evaluation task list contains one specification for each participant and condition, for $18 \times 4 = 72$ specifications. An optional `gpt-4o-mini` judge can be run as a separate phase, but it is not used in the reported rule-based panel.

Generalize benchmark. The Generalize split is generated by a separate, OOD-by-design pipeline that combines three rule-based families with an LLM verification pass that filters out tasks the KB could have answered as Replay. The first family, *transfer*, draws on a participant's consolidated **Inferred Knowledge** to identify "observed action a on object o_{obs}" pairs (e.g. P01 has been observed cutting mushrooms three times), then substitutes a target object o_{tgt} that the same participant has been observed but never in combination with a (e.g. `cutting potato`); the task query asks the agent to apply the skill to the substituted target.

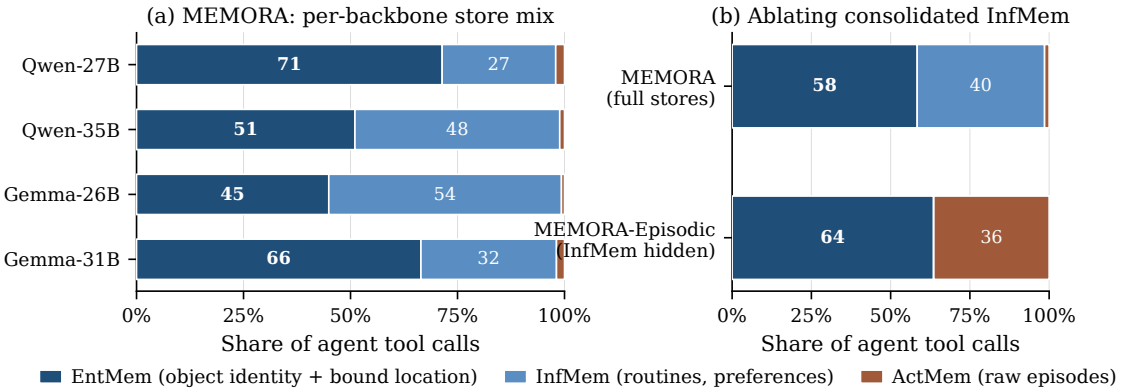


Figure 16: **Per-store contribution of agent tool calls during planning** (1,440 runs = 4 backbones × (207 Replay +153 Generalize tasks)). **(a)** Under the main MEMORA configuration, every backbone draws on multiple typed stores, but with markedly different mixtures: Gemma4-26B-A4B-it routes 54% of its tool budget to **Inferred Knowledge** (routines, preferences), while Qwen3.6-27B routes 71% to **Entity Memory** (object identity). A single-store substrate may fit these backbone families unevenly. **(b)** In the MEMORA-Episodic ablation, agents query **Entity Memory** and **Activity Memory** only (`search_objects`, `get_object_history`, `search_activities`); those queries are *not* eliminated — 35% of all calls migrate to **Activity Memory** episode search, showing that removing consolidated routines redirects substantial tool use to episodic activity search. **Environment Memory** is omitted from the figure because spatial information is bound to object records in the planning prompt (every `search_objects` return includes `location` attributes), so a separate environment-search tool is not advertised.

Table 17: Planning tool-call counts by typed store and configuration. *MEMORA-Episodic* exposes **Entity Memory** and **Activity Memory** tools only; full *MEMORA* additionally provides `get_routine_skill` and `get_preferences` over consolidated **Inferred Knowledge**. Percentages are normalized within each variant (7,163 vs. 8,331 total calls); tools with <1% usage in both variants are omitted.

Store	Tool	MEMORA		MEMORA-Episodic	
		calls	%	calls	%
EntMem	`search_objects`	4180	58.4	5103	61.3
InfMem	`get_routine_skill`	2530	35.3	0	0.0
	`get_preferences`	347	4.8	0	0.0
ActMem	`search_activities`	103	1.4	2912	35.0
Unified	`search`	3	0.0	297	3.6

The second family, *composition*, samples a high-level goal that is *not* present in `routine_skills` but whose sub-procedures individually are (e.g. `prepare a vegetable stir-fry dish` decomposes into operations such as heating a pan, picking up vegetables, and stirring, each of which appears as a routine but never as a parent skill). The third family, *fully-novel*, asks for plans whose surface goal has neither a matching routine nor a matching sub-procedure decomposition in the KB. An LLM verifier (`gpt-4o-mini`) is used *at generation only*: it writes a 5-step ground-truth plan for each candidate task and scores it on five binary criteria (novelty relative to KB routines, sub-procedure grounding, object grounding, physical correctness, and step-quality coherence); tasks that fail any criterion are dropped. The released split contains 153 verified tasks across the 18-PID panel. At evaluation time, the rule-based panel substitutes the task-level *PlanFid_GT* (verb-LCS against `ground_truth_steps`) for the KB-routine reference, because a Jaccard-matched nearest KB routine for a novel goal is only lexically close and penalizes agents that compose correctly; the metric router (`scripts/compute_rule_metrics.py`) auto-detects Generalize tasks by their `denovo_task_id` prefix and records the chosen reference in a per-task `metric_source` field.

C.6 Object-grounding diagnostic

We separately report object grounding as a diagnostic because no token-overlap definition of ob-

ject grounding cleanly captures what memory contributes to robot-followable planning. This subsection (i) defines four candidate object-grounding metrics, (ii) reports them across all six completed (backbone, benchmark) cells, and (iii) discusses why we ultimately exclude object grounding from RGP while still tracking it.

Four candidate definitions. Let P be the set of distinct kitchen-noun stems extracted from a generated plan, G the corresponding set for the ground-truth plan, and K the participant's KB-vocabulary noun set (§C). We consider:

$$\text{KBGround}(P, K) = |P \cap K|/|P|, \tag{8}$$

$$\text{ObjGT-R}(P, G) = |P \cap G|/|G|, \tag{9}$$

$$\text{ObjGT-P}(P, G) = |P \cap G|/|P|, \tag{10}$$

$$\text{ObjGT-F1} = \frac{2\,\text{ObjGT-P}\,\text{ObjGT-R}}{\text{ObjGT-P} + \text{ObjGT-R}}, \tag{11}$$

$$\text{GR}(P, K, G) = \frac{|P \cap K \cap G|}{|K \cap G|}, \tag{12}$$

$$\text{GP}(P, K, G) = \frac{|P \cap K \cap G|}{|P|}, \tag{13}$$

$$\text{GF1} = \frac{2\,\text{GP}\,\text{GR}}{\text{GP} + \text{GR}}. \tag{14}$$

KBGround answers "does the plan mention things known to the participant?" ObjGT-{R,P,F1} answers "does the plan mention the same objects as the GT plan?" GR / GP / GF1 (Grounded Recall / Precision / F1) answer the more robot-relevant question "does the plan mention the right task objects *and* are those objects findable in the participant's kitchen?"

Results across all completed cells. Table 18 reports the four grounding metrics for every (backbone, benchmark) cell. The pattern is consistent across cells: chunk-based retrieval baselines win KBGround by a wide margin (their plans inherit verbatim text from the chunks used to build the KB), and the winner under ObjGT-F1 or GF1 is typically Flat-1D or Graph-2D rather than MEMORA.

The chunk-copying signature: KeyObj and KBGround disagree on baselines. The contrast between KeyObj (in RGP) and KBGround (this diagnostic) is informative. On Qwen3.6-35B-A3B / Generalize, Flat-1D (raw) scores KBGround $=$ $.920$ but KeyObj $=$ $.570$; MEMORA scores KBGround $=$ $.883$ but KeyObj $=$ $.650$ — the chunk-copying signature. The Pearson correlation between KeyObj and KBGround at the per-task level is essentially zero on this cell (range $r{=}-.116$ to $+.220$ across conditions), confirming that the two metrics measure complementary properties: KeyObj rewards plans that name the per-task gold objects, while KBGround rewards plans that mention any KB-known noun. Folding both into a single composite would (i) double-weight "object naming" relative to procedure / preferences, and (ii) reward chunk-copying baselines on the KBGround side without rewarding task-relevant retrieval. RGP therefore keeps only KeyObj.

Why no token-overlap metric cleanly favors MEMORA. Across all four grounding-metric variants, MEMORA wins outright on at most one of six cells (Gemma-4-31B-it / Replay, ObjGT-F1 and GF1 tied at $+.001$). Three structural reasons explain this: (1) The Generalize task query directly names the key objects ("Help P01 *cut the cucumber*"), so any LLM with kitchen prior — including the No-Memory parametric baseline — produces a plan whose nouns largely match the GT plan; this is a verbosity confound that ObjGT-R rewards (No Memory wins ObjGT-R on every Generalize cell). (2) Flat-1D copies EPIC narrations verbatim, so its plans inherit both KB-membership and GT-noun-membership "for free" on Replay; Graph-2D injects graph-resolved entities sparsely and wins ObjGT-Precision by being terse. (3) MEMORA paraphrases retrieved objects into natural-language plan steps (e.g., "the green cucumber that P01 keeps on the counter") and adds procedural elaboration, both of which dilute token-overlap scores even when the underlying entity is correct. None of these metrics tests whether the *specific instance* named is correct (which cucumber, on which counter); answering that question requires execution-grounded evaluation in simulation or on a physical platform, which we leave to future work.

Table 18: Object-grounding diagnostic across all completed (backbone, benchmark) cells. For each cell we report the value for the best non-MEMORA baseline ("best base") and for the best MEMORA variant (over MEMORA, MEMORA-Episodic), and the gap Δ (MEMORA $-$ best base, positive favors MEMORA). ObjGT-F1 = GT-plan noun F1; GF1 = Grounded F1 (KB $\cap$ GT). MEMORA does not consistently win on any token-overlap grounding metric; the chunk-based retrieval advantage is robust across metric variants.

Backbone	**Bench**	**KBGround**		**ObjGT-F1**		**GF1 (KB∩GT)**	
		best base / MEM	Δ	best base / MEM	Δ	best base / MEM	Δ
Gemma-4-26B-A4B-it *(MoE, 4B active)*	Replay	.912 / .883	$-.029$	.481 / .482	$+.001$	.477 / .461	$-.016$
	Generalize	.916 / .885	$-.031$	.663 / .610	$-.053$	.629 / .579	$-.050$
Gemma-4-31B-it *(dense)*	Replay	.904 / .901	$-.003$	.490 / .491	$+.001$	.490 / .491	$+.001$
	Generalize	.918 / .905	$-.013$	.679 / .645	$-.034$	.638 / .616	$-.022$
Qwen3.6-35B-A3B *(MoE, reasoning)*	Replay	.909 / .883	$-.026$	.491 / .483	$-.008$	.494 / .452	$-.042$
	Generalize	.925 / .892	$-.033$	.622 / .598	$-.024$	.608 / .566	$-.042$

D Physical Robot Demonstration

This appendix documents the qualitative robot case study summarized in Figure 4 and referenced in Section 5.4; Figures 18 and 19 provide the complete traces. The purpose is not to introduce a separate robot benchmark, but to check whether the object-grounding errors measured by MEMORA-Planning remain visible when a memory-conditioned instruction is executed in a physical scene. The demonstration is a transfer setting: the memory is formed from a human demonstrator's egocentric video, then retrieved as procedural and object-grounding guidance for a robot acting in the same environment. We therefore report the setup, task design, success criterion, and video traces for two memory-grounded robot tasks.

D.1 Setup and Protocol

Platform and scene. The demonstration uses a Unitree G1 operating in a kitchen-like tabletop scene with visually similar distractor objects. Before robot execution, the human demonstrator records two egocentric videos in the same scene: one for a drink-preparation routine and one for a breakfast routine. These videos are processed before robot execution, so the robot plan is conditioned on memory formed from human first-person experience rather than on direct access to the demonstration video at execution time.

Task design. Each instruction is intentionally underspecified unless the system remembers the participant's prior choices. **Task A: Prepare Drink** tests preference and grounding: the robot must choose bottled Coca-Cola over a yellow Fanta pineapple can, and must choose the demonstrator's orange plastic cup among cups with other colors. **Task B: Breakfast** tests preference, grounding, and sequence: the robot must choose the yellow bowl rather than a gray bowl, place Cinnamon Tea rather than Lipton Black Tea into the bowl, place the yellow spoon rather than a metal spoon into the bowl, and then take the bread. The intended test is whether a memory system can transfer participant-specific human choices into a robot plan that resolves object identity, attributes, preferences, and temporal order, rather than following a generic task script.

Evidence. The demonstration consists of four videos: the human egocentric demonstration for Task A, the human egocentric demonstration for Task B, the Unitree G1 execution of Task A from MEMORA's plan, and the Unitree G1 execution of Task B from MEMORA's plan. The two robot videos provide the qualitative success traces reported below.

Execution interface. MEMORA produces a language-level plan for each participant-specific instruction. The generated plan is then converted into grasp and placement primitives by a fixed rule-based low-level controller. This design intentionally holds motor execution fixed, so the demonstration focuses on memory-grounded language planning rather than learned robot-control differences.

D.2 Results and Traces

Table 19 records the task-level plan and success criterion, and Figures 18 and 19 show the full visual traces used to construct the compact main-paper figure. Each panel includes the human egocentric demonstration and the corresponding Unitree G1 execution, making explicit which observed prefer-

Task	Distractors / ambiguity	Generated MEMORA plan used for execution	Success criterion
PREPARE DRINK	Fanta and other cups are visible, so the instruction alone does not specify the participant's drink setup.	(1) retrieve the orange cup; (2) retrieve the Coca-Cola bottle; (3) position the orange cup with the Coca-Cola bottle as demonstrated.	The robot selects Coca-Cola rather than Fanta and the orange cup rather than other cups.
BREAKFAST	Multiple bowls, tea items, utensils, and side items are present, so both choices and order depend on prior experience.	(1) select the yellow bowl; (2) add Cinnamon Tea; (3) add the yellow spoon; (4) bring bread.	The robot selects the remembered breakfast entities and preserves the demonstrated assembly order.

Table 19: **Robot-transfer task specification and generated plans.** The plans are language-level MEMORA outputs used to drive a fixed rule-based controller.

ences and object choices are transferred into robot behavior.

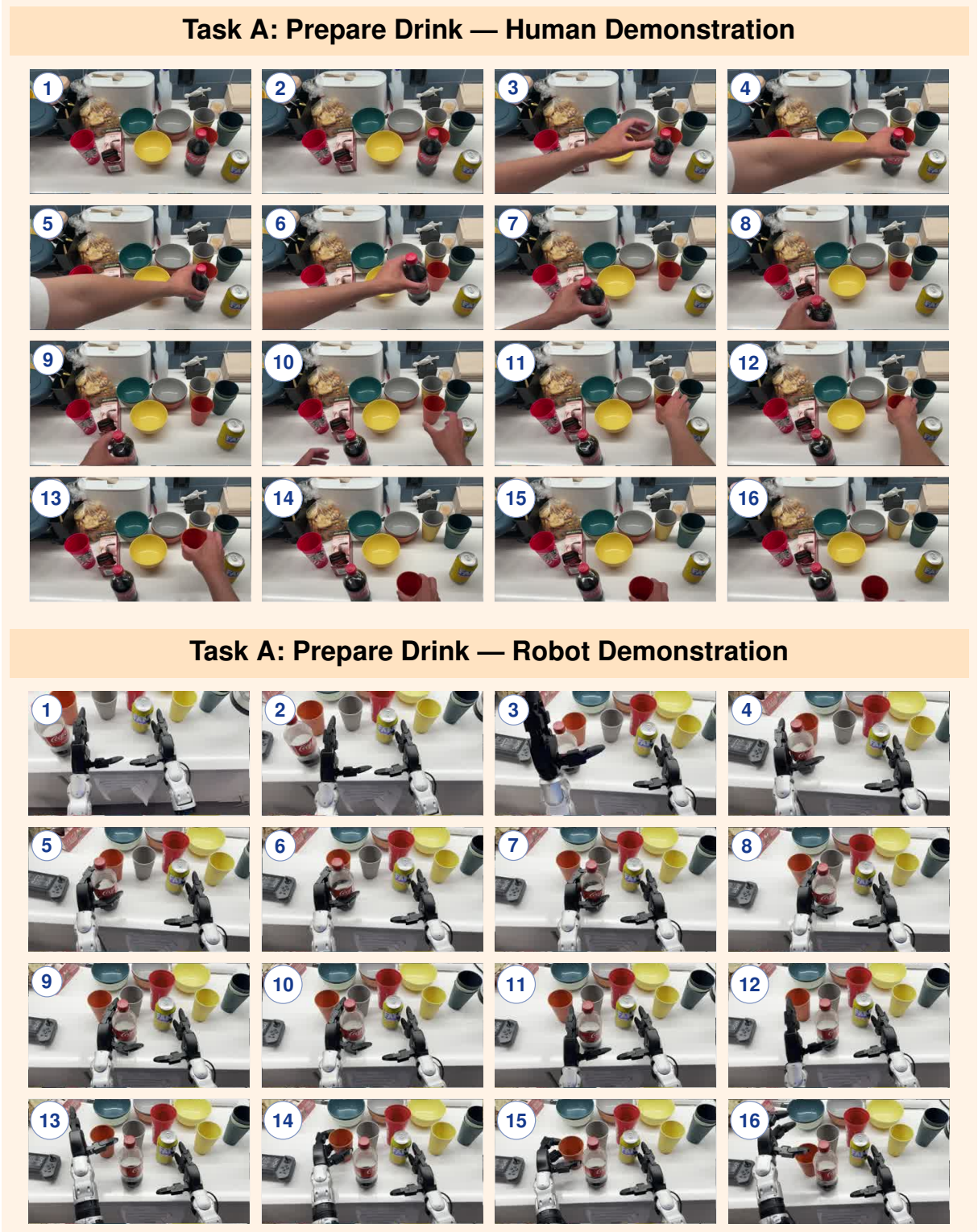


Figure 18: **Task A: Prepare Drink (full trace).** The contact sheet shows the human egocentric demonstration and the Unitree G1 execution guided by MEMORA's memory-grounded language plan and a fixed rule-based controller. This task tests preference and object grounding: the robot must select bottled Coca-Cola over a visually salient Fanta can and the demonstrator's orange plastic cup among multiple colored cups.

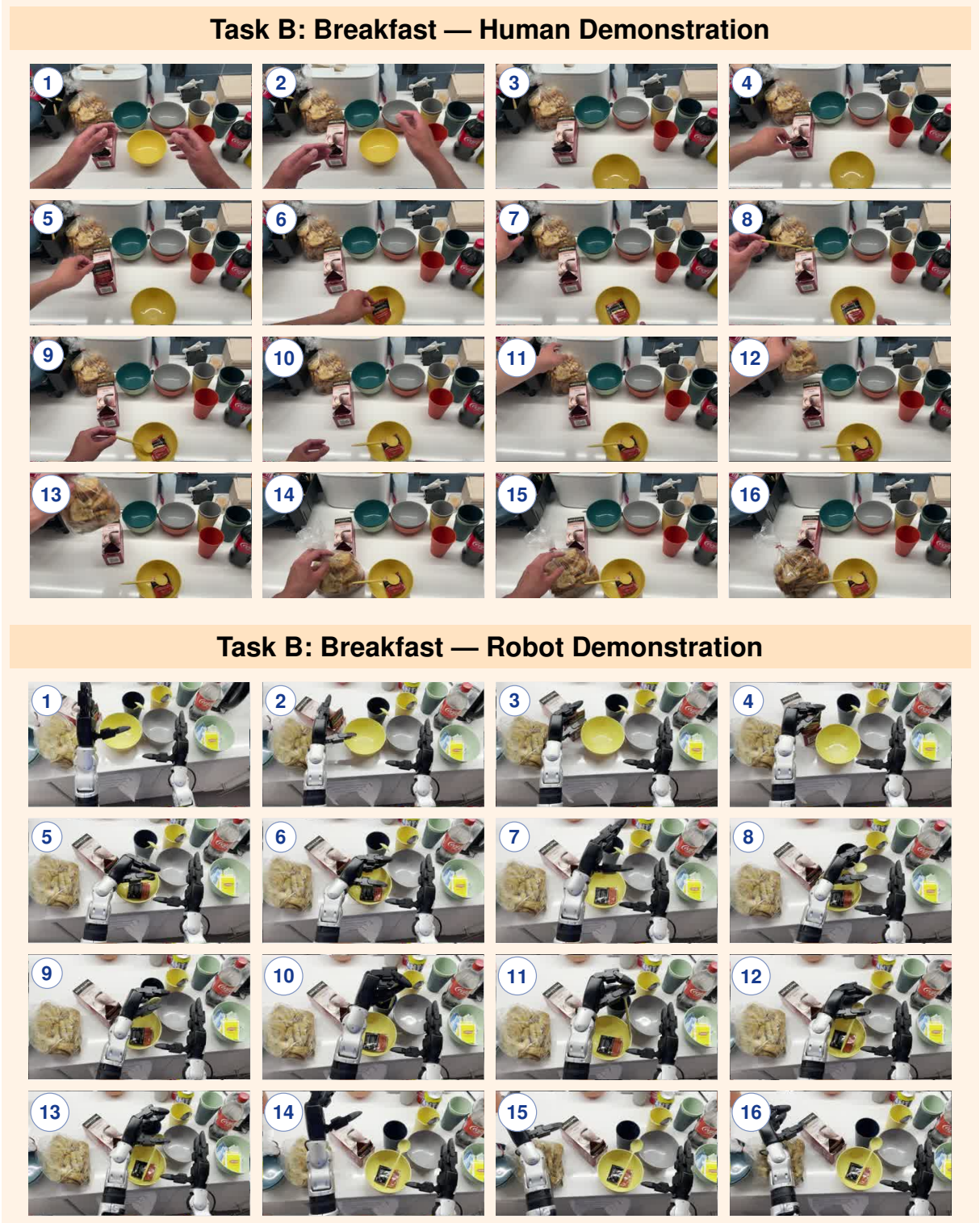


Figure 19: **Task B: Breakfast (full trace).** The contact sheet shows the human egocentric demonstration and the Unitree G1 execution guided by MEMORA's memory-grounded language plan and a fixed rule-based controller. This task adds ordered procedural execution: the robot must select the yellow bowl, Cinnamon Tea, yellow spoon, and bread in the remembered breakfast sequence.

Both tasks succeed on the first attempt (2/2): Task A selects bottled Coca-Cola and the orange plastic cup; Task B selects the yellow bowl, Cinnamon Tea, yellow spoon, and bread in the remembered order.

Limitations. This demonstration should be interpreted as a deployment-oriented sanity check. It involves two tasks in one physical scene and is not intended to establish statistical robot generalization. Its role is to make the planning metric concrete by showing that memory formed from human egocentric observation can guide physical object selection and sequencing in a robot execution setting.

Beyond the fixed-controller demonstration. The rule-based Unitree G1 controller used above is one simple instantiation of a broader real-robot stack; it is not a requirement imposed by MEMORA. MEMORA supplies a high-level goal or plan grounded in participant-specific memory, which can then be executed by a conventional feedback controller, a learned state- or vision-conditioned policy such as Diffusion Policy (Chi et al., 2023), or a general vision–language–action policy such as RT-2 or π_0 (Brohan et al., 2023; Black et al., 2026). When adaptation to a target robot or task is needed, parameter-efficient updates such as LoRA can specialize a pretrained policy (Hu et al., 2022), while online interaction offers an additional route for refinement: Policy Decorator adds a model-agnostic residual policy to a fixed imitation policy through controlled online exploration (Yuan et al., 2025), whereas RL Token (RLT) exposes a compact representation from a pretrained vision–language–action model to a lightweight actor–critic for sample-efficient real-robot online reinforcement learning (Xu et al., 2026). For long-horizon execution, a progress monitor can assess whether each intended state transition has occurred before the system advances or replans, using vision–language-model state grounding and replanning (Skreta et al., 2024), milestone-based progress monitoring and recovery (Dai et al., 2026), or action-conditioned world-model verification (Liu et al., 2026); human confirmation provides a conservative fallback when automated verification is unreliable. Under this decomposition, MEMORA provides persistent participant-specific context and high-level grounding, the downstream policy realizes the selected action, and the monitor closes the loop by deciding whether to continue, replan, or request intervention. These are practical pathways for broader evaluation rather than claims established by our two-task demonstration.

E Controlled Ablation Studies

Appendix E defines three orthogonal controlled ablations over the MEMORA memory-bank construction pipeline and its downstream answer agents. **Reference** uses the main-paper processor settings in Appendix F (processor contracts: Table 4); the released pipeline contains no closed-weight model in any role. **Cohort split.** We partition the 18-participant MEMORA-Bench panel into a *calibration* cohort and a *held-out evaluation* cohort, and run Axes 1–2 on the held-out cohort to avoid leakage of any processor tuning or prompt iteration into the ablation result:

- **Calibration cohort (P01–P04, 4 PIDs, ∼80 EPIC-KITCHENS extension videos, 746 EAM-QA questions = 27.0% of the full EAM-QA panel).** This subset was used during pipeline development to iterate perception- and memory-processor prompt templates and to fix the MEMORA pipeline; it is therefore not a clean test set for processor swaps.
- **Held-out ablation cohort (P06, P07, P09, P11, P12, P22, P23, P25, P26, P27, P28, P30, P35, P37 – 14 PIDs, ∼121 videos, 2,017 EAM-QA questions = 73.0% of the benchmark).** Memory-processor outputs for the three Axis-1/Axis-2 arms (Omni+Q30B, Omni+Q14B, VL+Q30B) were materialized in a single "rest" merge over exactly this PID set, so the comparison covers a fixed video population for every arm.

Isolation protocol: each study varies exactly one of {*online memory processor*, *perception processor (and audio)*, *answer agent*}; the other two axes remain at the main-paper settings in Appendix F and the **Reference** row of Table 20. Axes 1–2 use the **MEMORA-Episodic** memory bank (without offline **Inferred Knowledge** enrichment) so all three arms are compared on identical memory-processor outputs, isolating the perception→memory-encoding contribution before enrichment. All ablations reuse the processor prompt templates in Figure 7 and the planning prompt templates in Figure 14; only the ablated model, modality, or answer-agent backbone changes.

E.1 Master Condition Matrix

Axis 1: Online Memory Processor.

Question. How sensitive are downstream scores to the capacity of the *memory-editing* language model when segment perception is held constant?

Design. The Perception Processor is fixed at Qwen2.5-Omni-7B, and the *same* per-segment JSON is reused for both conditions; only the Online Memory Processor differs (A1-14B vs. Reference in Table 20). No **Inferred Knowledge** enrichment is applied to either tree, so the enrichment LLM is not part of the comparison. The answer agent is held fixed across conditions so EAM-QA isolates encoding quality rather than answerer capacity; we report results on two answer backbones (Gemma-4-26B-A4B-it and Qwen3.6-35B-A3B) over the same 14-participant cohort in Table 22, to verify that the encoding-level conclusion is not specific to a particular answerer.

Memory-bank shape under the A1 swap. Before turning to downstream accuracy, we compare the *shape* of the Entity Memory bank produced by the two memory processors on the same Omni-7B perception stream (14-PID held-out cohort). Table 21 reports, for each participant, the append-only entity-observation count (*No-edit*), the number of unique entities retained after editing (*MEMORA*), and the resulting compression ratio.

Two observations. First, the smaller editor's bank is *not* the result of tighter deduplication: the larger Q30B editor reaches a higher compression ratio on $13/14$ participants (median $20.6\times$ vs $18.2\times$; mean $+2.5\times$ paired difference). Second, the dominant axis of shrinkage is the *number of state changes recorded*: A1-14B commits 24.5% fewer turn-aligned entity observations to memory ($61{,}478 \rightarrow 46{,}438$) while keeping a similar number of unique entities (-5.0%, $2{,}800 \rightarrow 2{,}660$). The smaller editor therefore produces an *under-written* memory — it sees the same perception output but issues UPDATE and ADD less often — rather than a more compact one. This shape change is the mechanism behind the -5.8 point accuracy drop and the E-selection surge in Table 22 (Gemma block): temporal questions (SHABIT, SROUTINE) lose the most because they depend on cross-segment state-update density, not on the entity inventory.

Axis 2: Perception Processor and Audio.

Question. How much do perception backbone and **audio modality** contribute to encoded memory when the memory processor and answer agent are fixed?

Design. The Online Memory Processor is fixed at Qwen3-30B-A3B-Instruct-2507; we report results on two answer backbones in Table 22. **Reference** uses the Omni multimodal path with model-native audio so each 10 s segment is encoded jointly from video frames and the synchronized audio stream. The ablation swaps in **Qwen2.5-VL-7B-Instruct** (vision-only); because this backbone has *no* audio modality, A2-VL7 simultaneously tests (a) the multimodal $\rightarrow$ vision-only perception swap, and (b) the audio-absent condition for memory encoding, without requiring a separate audio-strip pass. Temporal preprocessing matches Appendix F: each 10 s segment is decoded to video frames at **2 FPS** and capped at $\leq$**24 frames** per segment; under Omni the native audio stream of the clip is additionally passed in, and under VL only the frame stack is passed in. The only thing that varies across conditions is therefore the perception-processor backbone and, by construction, its access to audio (A2-VL7 vs. Reference in Table 20).

Axis 3: Answer agent.

Question. How much do MEMORA-Bench and Planning scores depend on the *answer-agent* backbone when the encoded consolidated **Inferred Knowledge** bank is frozen to the main-paper pipeline output?

Design and reference to existing panels. Axis 3 is structurally identical to the cross-backbone evaluation already reported in the main supplementary material: all conditions load the same Reference consolidated **Inferred Knowledge** bank (Omni-7B perception, Qwen3-30B-A3B-Instruct-2507 memory) and only the answer agent changes. We therefore do not introduce a new Axis-3 table here; the four backbones {Qwen3.6-35B-A3B (Reference), Qwen3.6-27B, Gemma-4-26B-A4B-it, Gemma-4-31B} are scored on MEMORA-Bench EAM-QA in Table 10 and on MEMORA-Planning in Figure 15 (eight cells over Replay $\times$ Generalize $\times$ four backbones; per-axis detail in Table 16). The 18-participant cross-backbone Planning panel is a **superset** of the Axes 1–2 cohort (it includes the calibration PIDs and uses identical rule-based metrics and task prompts). For the Qwen3.6 family we disable extended thinking; Gemma models use

Table 20: Ablation conditions on the held-out 14-participant cohort (P06–P37) for Axes 1–2; Axis 3 is the cross-backbone MEMORA-Bench / Planning panel and is not duplicated here. Each row changes one axis relative to **Reference**; all other components match the **Reference** settings in Appendix F. Axes 1–2 are evaluated on the MEMORA-Episodic memory bank (without **Inferred Knowledge** enrichment) so the comparison is confined to the memory-processor outputs of the ablated pipelines. A2-VL7 is jointly a perception-backbone swap *and* an audio-off condition, because `Qwen2.5-VL-7B-Instruct` is a vision-only model with no audio modality; the multimodal vs. vision-only contrast against Reference therefore covers "audio off" as a side effect of the backbone change.

ID	Ax.	Changed component	Held fixed
Reference	–	Main-paper Reference: Qwen2.5-Omni-7B perception + Qwen3-30B-A3B-Instruct-2507 online memory processor	Main-paper perception, memory editing, and answer-agent settings; MEMORA-Episodic memory bank (without **Inferred Knowledge** enrichment).
A1-14B	1	Qwen3-14B online memory processor	Qwen2.5-Omni-7B perception; results reported on Gemma-4-26B-A4B-it and Qwen3.6-35B-A3B answer agents.
A2-VL7	2	Qwen2.5-VL-7B-Instruct vision-only perception (audio absent by construction)	Qwen3-30B-A3B-Instruct-2507 online memory processor; results reported on Gemma-4-26B-A4B-it and Qwen3.6-35B-A3B answer agents.
Axis-3 panel	3	Qwen3.6-35B-A3B, Qwen3.6-27B, Gemma-4-26B-A4B-it, and Gemma-4-31B-it answer agents	Fixed Reference consolidated **Inferred Knowledge** bank; reported in Table 10 and Figure 15 (per-axis detail in Table 16).

the project tool-call parsers without a standardized think mode.

Sanity check: closed-weight enrichment alternative. The main-paper pipeline performs **Inferred Knowledge** enrichment with Qwen3.6-35B-A3B, so the released system contains no closed-weight model. For completeness we also produce an alternative consolidated **Inferred Knowledge** bank using `gpt-4o-mini`, holding every other component (perception, online memory processor, retrieval encoder, answer agent) fixed. On the two backbones where the full cross-bank evaluation has finished, the two enrichment LLMs yield statistically equivalent EAM-QA accuracy: Gemma-4-26B-A4B-it scores 35.76% with the Qwen3.6 bank and 35.92% with the `gpt-4o-mini` bank ($\Delta = -0.16$ points on a triple-paired subset of N=1,843 questions); Gemma-4-31B scores 36.4% with the Qwen3.6 bank and 35.8% with the `gpt-4o-mini` bank (Δ=+0.6 points, the only cell where the open-weight bank *exceeds* the closed-weight one). Both gaps lie inside the typical seed/condition variance of EAM-QA at this N, so we conclude that the open-weight enrichment used in the main-paper pipeline does not sacrifice headline accuracy and that the entire MEMORA pipeline is reproducible without any closed-weight LLM.

E.2 Evaluation Protocol

Axes 1–2 are scored on the MEMORA-Bench EAM-QA harness over the held-out 14-participant cohort (P06–P37, 2,017 questions = 73.0% of the full EAM-QA panel) using the MEMORA-Episodic memory bank (without **Inferred Knowledge** enrichment), so perception- and memory-encoding effects are not confounded with **Inferred Knowledge** enrichment. Metrics are overall accuracy over five options (A–E, with E denoting information not available) and per-type accuracy on SPREF/SROUTINE/SHABIT/ERECALL. Axis 3 is read off of the existing cross-backbone panels (Table 10, Figure 15) without re-running.

E.3 Results

Axes 1–2 results on two answer backbones (14 PIDs, 2,017 EAM-QA questions). Table 22 reports overall accuracy, E-selection rate, and per-type accuracy for the three arms on the MEMORA-Episodic memory bank, scored on two answer backbones to verify that the encoding-level conclusion is not specific to a particular answerer.

Table 21: Axis 1 memory-bank shape on the 14-participant held-out cohort (Qwen2.5-Omni-7B perception fixed). *No-edit* counts turn-aligned entity observations that would be stored under an append-only policy; *MEMORA* counts unique entities retained in the edited `object_registry`. *Ratio* is No-edit / MEMORA, the per-participant Entity Memory compression factor. A1-14B commits fewer state-history events than Reference (lower No-edit) while keeping a comparable entity inventory (similar MEMORA).

	Reference (Qwen3-30B-A3B-Instruct-2507)			**A1-14B (Qwen3-14B)**		
PID	**No-edit**	**MEMORA**	**Ratio**	**No-edit**	**MEMORA**	**Ratio**
P06	4,182	263	15.9×	3,614	263	13.7×
P07	2,402	148	16.2×	1,977	156	12.7×
P09	818	87	9.4×	833	98	8.5×
P11	1,849	134	13.8×	1,055	129	8.2×
P12	3,409	112	30.4×	1,813	79	22.9×
P22	7,334	382	19.2×	6,170	377	16.4×
P23	3,942	88	44.8×	3,535	89	39.7×
P25	5,477	137	40.0×	2,907	142	20.5×
P26	4,193	246	17.0×	1,774	195	9.1×
P27	4,609	210	21.9×	3,117	152	20.5×
P28	5,099	299	17.1×	3,484	276	12.6×
P30	6,374	282	22.6×	6,134	306	20.0×
P35	9,259	311	29.8×	7,925	321	24.7×
P37	2,531	101	25.1×	2,100	77	27.3×
Total	**61,478**	**2,800**	**20.6× med.**	**46,438**	**2,660**	**18.2× med.**

Findings. On the same answer agent (Gemma-4-26B-A4B-it) and the same 14-PID MEMORA-Episodic bank: (i) **Memory-processor capacity dominates**: dropping the online memory processor from Qwen3-30B-A3B-Instruct-2507 to Qwen3-14B costs -5.8 points of overall accuracy and raises the E-selection rate from 42.8% to 62.3%, indicating that the smaller model produces memory-bank records for which the answer agent more often selects information not available. Table 21 traces this to the *shape* of the memory bank rather than to over-deduplication: A1-14B writes 24.5% fewer turn-aligned entity observations into Entity Memory while keeping a comparable entity inventory (-5.0%), so the underlying state-change record is sparser rather than more compact. (ii) **Perception backbone matters less**: swapping the Omni multimodal path for vision-only Qwen2.5-VL-7B costs -2.7 points overall with an essentially unchanged E-selection rate ($+1.3$ points), suggesting that, at MEMORA-Episodic granularity, audio-aware perception provides a smaller marginal contribution than memory-editor capacity. (iii) **Type-conditional behavior is consistent with the encoding interpretation**: A1-14B preserves ERecall ($+1.3$ points) but loses the most on SPref / SRoutine / SHabit ($-6.4/-8.3/-6.8$ points), which are the types that depend most on cross-segment habit/preference consolidation; A2-VL7 instead spreads its loss approximately evenly across all four types ($-2.2/-3.1/-2.7/-2.8$ points).

Cross-backbone confirmation (Qwen3.6-35B-A3B answer agent, full 14 PIDs). On the same 14-PID held-out cohort and the identical MEMORA-Episodic memory banks, swapping the answer agent from Gemma-4-26B-A4B-it to the larger reasoning model Qwen3.6-35B-A3B (Table 22, Qwen block) preserves the sign and ranking of both effects: memory-processor ablation costs -3.3 points overall versus -1.3 points for perception ablation, so memory-processor capacity remains the larger contributor across answer backbones. The type-conditional pattern is also preserved: A1-14B loses on the cross-segment types (-6.6 on SRoutine, -6.2 on SHabit) while ERecall is essentially unchanged ($+1.0$), and A2-VL7 spreads its loss more evenly across types. Two backbone-specific nuances are visible. First, Qwen3.6-35B-A3B is a more committed answerer: its Reference E-rate is 30.2% versus 42.8% for Gemma-4-26B-A4B-it, and the A1-14B E-rate increase is correspondingly smaller ($+5.0$ points vs $+19.5$ points), so the under-written memory hurts accuracy through more wrong commits rather than through more E selections. Second, the absolute Δ for A1-14B is smaller in magnitude on the larger reasoning backbone (-3.3 vs -5.8 points), consistent with a stronger answerer extracting more signal from a sparser memory record but not closing the gap. The encoding-level conclusion — memory-

Table 22: Axes 1–2 results on the 14-participant cohort, MEMORA-Episodic memory bank (without **Inferred Knowledge** enrichment), reported on two fixed answer backbones. "Δ vs Reference" is the overall-accuracy gap relative to **Reference** (Qwen2.5-Omni-7B perception + Qwen3-30B-A3B-Instruct-2507 online memory processor) within each backbone block. Per-type columns are overall accuracy on the eligible EAM-QA subset for that type. The sign and ranking of both ablation effects (A1-14B costs more than A2-VL7, loss concentrated on cross-segment types) are preserved across answer backbones.

Condition	Acc.	E-rate	Δ vs Reference	SPref	SRoutine	SHabit	ERecall
Answer backbone: Gemma-4-26B-A4B-it							
Reference (Qwen2.5-Omni-7B / Qwen3-30B-A3B-Instruct-2507)	**46.7**	42.8	—	47.4	49.9	43.4	45.1
A1-14B (Qwen2.5-Omni-7B / Qwen3-14B)	40.9	62.3	−5.8	41.0	41.6	36.6	46.4
A2-VL7 (Qwen2.5-VL-7B-Instruct / Qwen3-30B-A3B-Instruct-2507)	44.0	44.1	−2.7	45.2	46.8	40.7	42.3
Answer backbone: Qwen3.6-35B-A3B							
Reference (Qwen2.5-Omni-7B / Qwen3-30B-A3B-Instruct-2507)	**53.4**	30.2	—	52.0	60.8	47.3	53.9
A1-14B (Qwen2.5-Omni-7B / Qwen3-14B)	50.1	35.2	−3.3	51.6	54.2	41.1	54.9
A2-VL7 (Qwen2.5-VL-7B-Instruct / Qwen3-30B-A3B-Instruct-2507)	52.1	33.8	−1.3	53.5	57.9	45.2	50.8

processor capacity dominates over perception backbone — therefore replicates across both answer backbones we tested, and the Q3.6-35B-A3B block in Table 22 supersedes the earlier P11+P12 pilot reading.

Qualitative side-by-side comparison. Figure 20 contrasts the three arms on two illustrative EAM-QA items, picked to expose the two ablation-specific failure modes implied by the aggregate accuracy and E-selection numbers in Table 22 (Gemma block). On both items, Reference (Omni + Qwen3-30B-A3B-Instruct-2507) recovers the gold letter; the memory-ablation arm A1-14B (Omni + Qwen3-14B) selects "E" (information not available), consistent with the +19.5-point E-rate increase; and the perception-ablation arm A2-VL7 (VL + Qwen3-30B-A3B-Instruct-2507) confidently picks a visually plausible but *habitually* wrong option, consistent with a vision-only backbone that lacks the audio-aware multimodal context that the Omni-encoded memory bank otherwise propagates into the memory-processor records.

Example 1. P06 – type SPREF; gold answer **D**.
Q: After using the knife-maintenance tool, where does P06 seem to keep it?

A Leaves it by the sink
B Sets it on the counter
C Stores it in a cupboard
D Places it in a drawer *(gold)*
E The information is not available based on the given context

Arm	Pred.	Outcome
Reference (Omni + Q30B)	D	correct – consolidated habit "puts away after use" recovered from the multimodal memory bank.
A1-14B (Omni + Q14B)	E	E selected – the smaller memory processor leaves the put-away action under-consolidated, so the answer agent selects information not available.
A2-VL7 (VL + Q30B)	B	wrong – the vision-only frames capture the tool resting on the counter *during* use; without the surrounding audio/routine context that Omni provides, the model latches onto that visual cue and commits to the wrong letter.

Example 2. P07 – type SHABIT; gold answer **B**.
Q: When P07 opens the bag during this cooking sequence, what do they typically do next?

A Seal the bag and set it beside the stove
B Take the omelette and place it onto the pan *(gold)*
C Move the bag aside and pick up a spoon
D Empty the bag into a bowl on the counter
E The information is not available based on the given context

Arm	Pred.	Outcome
Reference (Omni + Q30B)	B	correct – the consolidated habit "bag-open → omelette-to-pan" is preserved across cooking sessions.
A1-14B (Omni + Q14B)	E	E selected – the weaker memory processor under-records the cross-segment transition, and the answer agent again selects information not available.
A2-VL7 (VL + Q30B)	A	wrong – without audio cues (sizzle, stove sounds), the vision-only encoder reads the closing/sealing motion of the bag as the most salient next action and commits to a plausible-looking but incorrect letter.

Figure 20: **Qualitative ablation contrast on two EAM-QA items** (Gemma-4-26B-A4B-it answer agent; MEMORA-Episodic memory bank; held-out 14-PID cohort). Both items illustrate the two failure modes inferred from the aggregate numbers in Table 22 (Gemma block): *(i)* the memory-ablation arm (A1-14B) loses the contentful answer and selects E more often; *(ii)* the perception-ablation arm (A2-VL7) keeps committing to an A–D option but the answer is grounded in single-segment visual evidence rather than cross-segment audio-aware consolidation, producing a confidently wrong letter.

F Implementation details

This section consolidates the model registry, inference stack, perception preprocessing, and retrieval/agent-loop settings that are shared across benchmark generation (Appendix A), Embodied Memory evaluation (Appendix B), Planning evaluation (Appendix C), and the controlled ablations (Appendix E). Unless stated otherwise, every condition that involves a given role (perception processor, online memory processor, **Inferred Knowledge** enrichment, retrieval encoder, or answer agent) uses the values reported here, so that cross-condition comparisons are not confounded by undocumented hyperparameter drift.

Compute and inference stack. All open-weight models are served with vLLM 0.19.1 inside a Singularity container so that the dependency graph (CUDA / PyTorch / Transformers / sentence-transformers) is identical across runs. Two GPU classes are used: 80 GB H100 nodes and 80 GB A100 SXM4 nodes; the A100 timing node uses two AMD EPYC 7543 32-core CPUs (64 hardware threads). The perception processors (Qwen2.5-Omni-7B, Qwen2.5-VL-7B-Instruct) are served single-GPU (TP$=1$, `max_model_len` $\in \{32\,768, 49\,152\}$). The online memory processors (Qwen3-30B-A3B-Instruct-2507, Qwen3-14B) are served with TP$=2$, `max_model_len`$=32\,768$, `gpu_memory_utilization`$=0.85$, and sampling temperature 0.6. All answer agents (Qwen3.6-27B, Qwen3.6-35B-A3B, Gemma-4-26B-A4B-it, Gemma-4-31B-it) use the same TP$=2$ stack and temperature; the cross-backbone EAM-QA panel uses `max_model_len`$=65\,536$, while Planning and the standard EAM-QA harness use `max_model_len`$=32\,768$. For Qwen3.x family chat templates we set `enable_thinking=false` so that the long "thinking" channel does not consume the 65 536-token budget that the ReAct loop reserves for tool returns; Gemma chat templates are used as-released. The **Inferred Knowledge** enrichment pass is performed by Qwen3.6-35B-A3B served on the same vLLM stack (TP$=2$, temperature 0.6), so the main-paper pipeline contains no closed-weight model in any role. Enrichment is run once per participant during memory-bank construction and its outputs are then frozen for all evaluation conditions. A closed-weight alternative (`gpt-4o-mini`) is benchmarked in Appendix E.1.

Licences, intended use, and public release. All artifacts used in this work are publicly released for academic research and we use them consistently with their intended use. The EPIC-KITCHENS-100 dataset (Damen et al., 2022) is distributed under the CC BY-NC 4.0 licence for non-commercial research on egocentric video understanding, which covers the embodied-memory benchmarking we perform. The Qwen2.5 / Qwen3 / Qwen3.6 model families (perception, memory editor, **Inferred Knowledge** enrichment, and answer-agent roles) are released by Alibaba under the Tongyi Qianwen / Apache-2.0 family of open-weight licences for research and downstream development. The Gemma model family (Gemma-4-26B-A4B-it, Gemma-4-31B-it; used as answer agents) is released under the Gemma Terms of Use for permitted academic and non-commercial use. `gpt-4o-mini` is accessed through OpenAI's API only as a closed-weight sanity check for the **Inferred Knowledge** enrichment slot (Appendix E.1); no headline result depends on it. We add no new recordings, identifiers, or personally identifying content beyond what is already present in EPIC-KITCHENS-100; participant identifiers follow the dataset's pseudonymous labels (`P01`, `P02`, ...). MEMORA pipeline source code and MEMORA-Bench task lists are available in the public MEMORA repository under the licences distributed with the release (Apache-2.0 for code; CC BY-NC 4.0 for derived task data, inherited from the upstream dataset).

Model registry. Table 23 lists the Hugging Face identifier (or API endpoint) for every model role used in the paper; processor input/output contracts for memory-bank construction appear separately in Table 4. A condition can be uniquely reconstructed as a triple (perception processor, online memory processor, answer agent); the **Inferred Knowledge** enrichment slot is either `Qwen/Qwen3.6-35B-A3B` (main-paper, consolidated **Inferred Knowledge** bank), `gpt-4o-mini` (closed-weight alternative, used only for the sanity check in Appendix E), or NONE (MEMORA-Episodic ablation bank without **Inferred Knowledge** enrichment).

Perception preprocessing. Each EPIC-KITCHENS video is split into non-overlapping 10-second segments; frames are decoded at 2 FPS and capped at 24 frames per segment, which matches the receptive horizon of the Qwen2.5-Omni / Qwen2.5-VL temporal encoder

Table 23: Model registry grouped by pipeline role. Total and active parameter counts coincide for dense models; for mixture-of-experts models, active denotes per-token compute.

Role	Model	Identifier	Total	Active
Perception processor (frame/audio to layered segments)				
Audio-visual	Qwen2.5-Omni-7B	`Qwen/Qwen2.5-Omni-7B`	7B	7B
Vision only	Qwen2.5-VL-7B-Instruct	`Qwen/Qwen2.5-VL-7B-Instruct`	7B	7B
Online memory processor (segments to typed memory records)				
Main	Qwen3-30B-A3B-Instruct-2507	`Qwen/Qwen3-30B-A3B-Instruct-2507`	30B	3B
Ablation	Qwen3-14B	`Qwen/Qwen3-14B`	14B	14B
Inferred Knowledge *enrichment (offline consolidation)*				
Main	Qwen3.6-35B-A3B	`Qwen/Qwen3.6-35B-A3B`	35B	3B
Alternative	GPT-4o-mini (API)	`openai/gpt-4o-mini`	—	—
Evaluation – answer agent (memory-conditioned QA / planning)				
MoE	Qwen3.6-35B-A3B	`Qwen/Qwen3.6-35B-A3B`	35B	3B
Dense	Qwen3.6-27B	`Qwen/Qwen3.6-27B`	27B	27B
MoE	Gemma-4-26B-A4B-it	`google/gemma-4-26B-A4B-it`	26B	4B
Dense	Gemma-4-31B-it	`google/gemma-4-31B-it`	31B	31B
Retrieval				
Encoder	E5-base-v2	`intfloat/e5-base-v2`	110M	110M

and keeps per-segment context length below `max_model_len`. For the audio-visual front-end (Qwen2.5-Omni-7B) the native audio stream of the segment is passed alongside the frames through the model's released audio tower; for the vision-only ablation (Qwen2.5-VL-7B-Instruct) the audio stream is dropped at decode time and only frames are presented. The perception processor outputs a JSON "layered segment" record per 10-second window (`action`, `objects`, `verbalised_audio` when available, and a free-form scene caption), which is then handed to the online memory processor.

Retrieval, dedup, and agent loop. At evaluation time the answer agent sees only an EAM-QA question (or a Planning request) plus a set of typed retrieval tools over the participant's memory bank. For EAM-QA, all memory-backed conditions expose `search` under the representation-specific prompt. Flat-1D exposes only this chronological-text search; Graph-2D and MEMORA additionally expose `get_object_history` and `get_state_at_time` over their respective representations. MEMORA's unified `search` can dispatch to object, activity, environment, and pattern evidence in the four typed stores. The history and state helpers are read tools over the loaded participant memory, not evidence that the released benchmark automatically rolls memory back to a question-specific time. All search tools use the same dense encoder (E5-base-v2) and return the top-k=10 records by cosine similarity, with a deduplication pass that collapses any pair of records whose embeddings have cosine similarity ≥ 0.95 to the higher-scoring one. The E5 retrieval encoder remains on GPU during evaluation; the reported memory-tool latency therefore includes GPU embedding time together with CPU-side orchestration, filtering, and vector-search overhead. The Embodied Memory ReAct loop allows at most 5 tool-call iterations per question; the Planning ReAct loop allows at most 8 (including a forced-answer final iteration), reflecting the longer intermediate-tool-use traces (`get_routine_skill` $\rightarrow$ `search_objects` $\rightarrow$ `search_activities` $\rightarrow$ `get_preferences`) that Planning queries require. Planning loads the planner once per task-list run through in-process vLLM rather than an OpenAI-compatible HTTP server, so weights stay resident across the full specification batch and tool-call parsing stays inside the evaluation process. Planning latency is measured after one-time model loading on the main Qwen3.6-35B-A3B stack served on A100/vLLM. In a timing run over 207 goals, end-to-end retrieval-and-planning takes 10.40 s on average, with most of the measured component time spent in answer-agent generation rather than memory access (Table 24). Because MEMORA forms memory offline before task execution and issues the language-level plan before low-level control, this measurement characterizes planning-time rather than motor-control latency. At Embodied Memory evaluation time, each task loads the fixed memory

Table 24: Planning latency on the main Qwen3.6-35B-A3B stack after one-time model loading (n=207 goals).

Component	Mean	Median	P90
End-to-end per goal	10.40 s	9.61 s	13.64 s
LLM generation	5.74 s	5.75 s	7.03 s
Memory-tool calls	1.06 s	0.78 s	2.40 s
Other overhead	3.59 s	2.73 s	6.42 s

bank of its declared target participant. The released headline questions contain no `ask_turn_id`, so the runner does not apply per-question state rollback. E-correct controls instead test unsupported-answer behavior by presenting donor-derived content options against a target-participant memory bank that contains no corresponding evidence.