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

Zihao Yu Xiu Yuan Chongjie Zhang
Washington University in St. Louis
{yu.zihao, xiu, chongjie}@wustl.edu
yuzihaowashu.github.io/MEMORA/

***Abstract*—Long-horizon robot planning requires more than predicting what actions will do next; it also requires memory of the embodied experience that makes future goals interpretable. People do not plan from the present scene alone: they draw on remembered places, object-state changes, prior procedures, and regularities revealed through repeated action. We formulate *Embodied Action Memory* (EAM) as the capability to form, maintain, and use such experience as a persistent memory state for later decisions. MEMORA realizes EAM with a formation–consolidation–retrieval lifecycle and four typed stores: Environment Memory, Entity Memory, Activity Memory, and Inferred Knowledge. Online editing maintains object identities and state histories as new observations arrive; offline consolidation abstracts repeated experience into reusable procedures and participant-specific regularities. MEMORA-Bench evaluates this lifecycle on 45 hours of EPIC-KITCHENS-100 extension video across 18 participants through memory-grounded planning, including previously unseen goals, and a complementary memory-assessment task. Across four open-weight language models, full MEMORA—combining editing, typed stores, and consolidation—achieves the strongest aggregate results among the evaluated memory conditions. It improves memory-assessment accuracy by up to 20.5 points over the strongest controlled baseline and improves out-of-distribution Robot-Grounded Plan score by up to 16.6% relative. A qualitative two-task robot deployment study further illustrates how memory-grounded language plans can interface with downstream control, while the overall results show that editable, consolidated memory can supply remembered context for robot planning.**

I. Introduction

Foundation-model robot planners can translate language goals, scene observations, and retrieved demonstrations into action sequences. Yet long-horizon planning in familiar environments also depends on memory formed from accumulated embodied experience. Appropriate action may depend on where a participant tends to keep tools, how objects have changed state, which procedures have appeared before, and what regularities past behavior reveals. For an embodied agent, such experience is not merely extra context; it is part of what makes a future goal interpretable for a particular person and environment.

Human planning reflects this dependence on memory. When preparing a meal, cleaning a workspace, or arranging tools, people use memory formed through observation and repeated action. Research in cognitive science separates episodic traces of particular events from semantic regularities abstracted across experience, with consolidation turning repeated episodes into reusable knowledge [44, 33, 43]. For embodied agents, this suggests a planning-facing memory capability beyond larger context windows: event evidence, changing state, and recurring procedures need to be maintained as experience unfolds and reused when new goals are posed. We call this capability **Embodied Action Memory (EAM)**.

Egocentric video is a natural source for EAM because it records experience from the actor's perspective. Across sessions, it captures a participant moving through familiar spaces, manipulating objects, and repeating everyday procedures. Recent egocentric-to-robot efforts use such first-person human video to scale imitation, dexterous manipulation, and cross-embodiment robot learning [18, 41, 58]. MEMORA studies a complementary use of the same signal. Instead of converting egocentric video directly into a policy, action prior, or robot-compatible trajectory, it converts accumulated experience into persistent semantic and procedural memory that helps a planner identify which routine, object, preference, or prior state is relevant. The challenge is to transform a continuous stream into memory that remains useful after individual episodes have passed, while still preserving experience-grounded context and temporal change.

This motivates a lifecycle view of memory. Classical cognitive architectures, differentiable memory agents, and recent LLM-agent memories all distinguish memory formation, maintenance, retrieval, and consolidation in different forms [4, 3, 19, 47, 13, 29, 36, 53, 10]. For egocentric embodied experience, the key question is how experience becomes memory as it unfolds. A useful memory system has to encode the perception stream into structured observations, revise the persistent state as new evidence arrives, consolidate repeated episodes into reusable regularities, and retrieve the right mixture of procedural and grounding evidence for planning. This lifecycle also has to respect the different continuities present in embodied experience: places persist, objects change state while retaining identity, actions unfold in temporal order, and routines emerge only across events. These requirements are difficult to satisfy by storing clips, flattening observations into text, or maintaining a single static entity graph alone [56, 28]. MEMORA therefore centers a lifecycle view: embodied experience becomes useful for planning when the perception stream is encoded, edited, consolidated, and retrieved as typed memory rather than treated as a passive archive.

MEMORA implements this lifecycle with typed memory stores and active maintenance mechanisms (Figure 1). Its

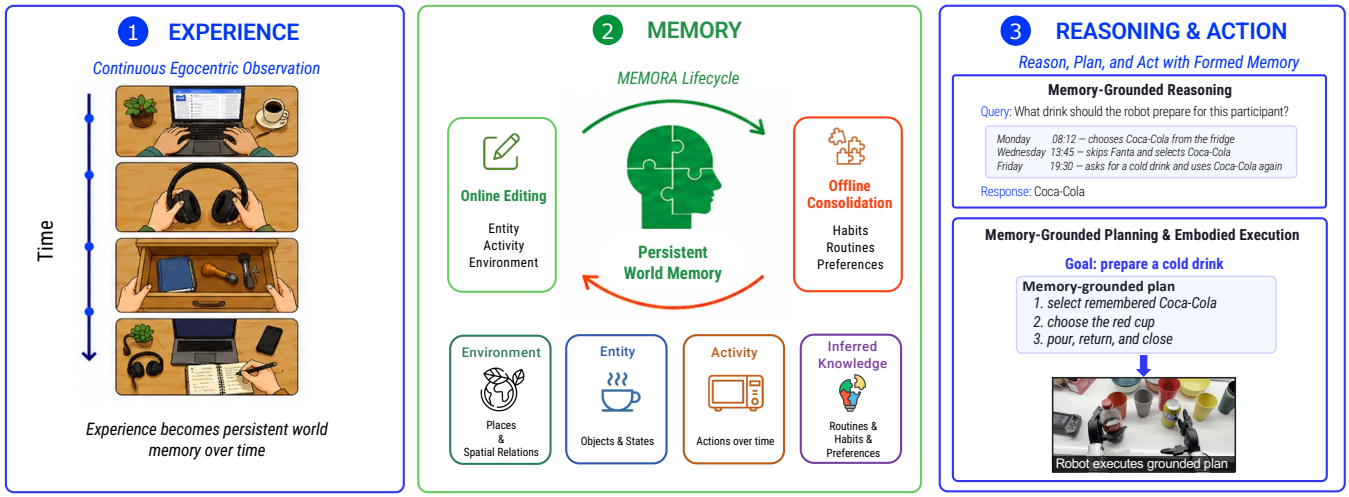

Fig. 1: **MEMORA studies embodied memory as a lifecycle from experience to action.** An agent observes continuous egocentric experience, forms typed memory online, and consolidates repeated experience into reusable routines, habits, and preferences. The resulting memory supports experience-grounded reasoning,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

memory state maintains four typed stores: **Environment Memory** for spatial context, **Entity Memory** for object identity and state, **Activity Memory** for temporally ordered action evidence, and **Inferred Knowledge** for regularities consolidated across repeated experience. Online, a Memory Editor actively maintains entity memory through ADD, UPDATE, DELETE, and NOOP operations, preserving object identities and state histories as new observations arrive. Offline, consolidation abstracts routines, habits, and preferences; at inference time, typed retrieval exposes both procedural regularities and physical grounding to the planner.

We evaluate EAM along two axes: whether memory preserves past experience and whether it supports future action. **MEMORA-Bench** contains 45 hours of EPIC-KITCHENS-100 extension video across 18 participants [2], paired with **MEMORA-Embodied Memory Assessment** and **MEMORA-Planning**. Planning includes **Replay** tasks grounded in observed workflows and **Generalize** tasks requiring transfer, composition, or new goals. Across four open-weight language models, full MEMORA achieves the strongest aggregate results among the evaluated memory conditions. It improves memory-assessment accuracy by up to 20.5 points over the strongest controlled baseline and improves out-of-distribution Robot-Grounded Plan score by up to 16.6% relative. A qualitative two-task physical-robot deployment check shows the resulting language-level plans grounding remembered action choices for downstream execution. This places MEMORA on the planning side of the egocentric-to-robot pipeline: policy-learning methods can learn how actions are executed, while MEMORA supplies remembered context for deciding and grounding what to do.

Contributions. (1) A formulation of **Embodied Action Memory (EAM)** (§III) as a memory substrate for embodied reasoning and planning, with a taxonomy separating spatial, entity-state, temporal-action, and consolidated-regularity continuities. (2) **MEMORA** (§IV), a framework that instantiates this dependency chain on real egocentric video through typed stores, active editing, offline consolidation, and typed retrieval. (3) **MEMORA-Bench**, a 45-hour evaluation suite pairing retrospective memory faithfulness with prospective memory-grounded planning. (4) A **controlled empirical study** isolating the contributions of online editing and offline consolidation and showing improvements in both memory assessment and planning, with the largest gains out of distribution.

II. Related Work

a) Foundation-model planning for embodied agents.: Language models and vision-language models now connect perception, reasoning, and action in embodied systems. Prompted planners, affordance-aware language models, code-generation approaches, and inner-monologue agents translate language goals and scene context into executable plans or skill calls [56, 1, 25, 16]. Vision–language–action models further learn policies that map multimodal observations and instructions directly to robot actions [6, 5]. MEMORA studies the memory layer that can condition such planners and policies: prior embodied experience is written, revised, consolidated, and queried when a later goal needs experience-grounded context.

b) Egocentric human video for robot learning.: Egocentric human video is a major data source for robot learning, including large-scale imitation, dexterous manipulation, vision–language–action pretraining, active-vision imitation, and methods that convert human demonstrations into robot-compatible observations or trajectories [18, 15, 50, 22, 26, 8, 58, 46, 27, 38, 55, 41]. These approaches typically use first-person video as demonstration data for learning policies, action representations, hand or head motion, active viewpoint control, or cross-embodiment alignment. MEMORA uses the same kind of signal for a different but compatible object of study: memory formation over the participant's places, entities, actions,

routines, and preferences. This makes egocentric policy learning and MEMORA naturally composable: one learns how actions can be executed on an embodiment, while the other supplies experience-grounded context for choosing and grounding what to do.

c) Embodied and spatial memory.: Several embodied agents maintain spatial, semantic, or 3D memory to support navigation, grounding, and long-horizon interaction. Recent systems build 3D scene memories, metric-aligned video memories, or multi-memory robotic agents for interactive environmental learning [51, 59, 21]. Such memories support localizing objects, grounding language in physical scenes, and acting over extended horizons. Recent robot-memory systems often accumulate an agent's own task trials, scene observations, or interaction history for later reuse, whereas MEMORA forms typed memory from longitudinal egocentric human experience and evaluates its use for both retrospective memory questions and prospective planning. MEMORA differs from geometric scene mapping and visuomotor memory: its units are typed records of embodied experience, including spatial context, object identity and state, action traces, and consolidated regularities. The resulting memory can be queried by planners or policies as semantic-procedural context, leaving metric prediction, control, and action generation to the corresponding robot modules.

d) Memory formation in language agents.: Agent memory has been modeled as parametric storage in model weights [57] or as explicit stores retrieved as documents, embeddings, summaries, graphs, or edited records [23, 60, 39, 40, 28, 7, 49]. Recent work further revisits the formation–consolidation–retrieval lifecycle for LLM agents, introducing explicit encoding and consolidation stages over text or simulated experience [29, 36, 53, 10]. MEMORA brings this lifecycle to real egocentric video. It is revised online through write-time active editing and organized by embodied continuities instead of a single transcript, summary, or graph. This design draws on cognitive distinctions between episodic, semantic, and consolidative memory [44, 33, 43, 37], but instantiates them as typed stores used by a planner.

e) Benchmarks for embodied memory.: Long-video QA benchmarks evaluate comprehension of video supplied at query time [11, 32, 24], while conversational-memory benchmarks isolate memory in text-only interactions [30, 48]. Egocentric datasets provide rich embodied activity, but they do not evaluate whether an agent can form memory across sessions and later use it for planning without receiving the relevant video at test time [2, 12]. MEMORA-Bench targets this experience-to-memory setting: questions and planning goals are grounded in experience-dependent evidence remembered across sessions.

III. Problem Formulation: Embodied Action Memory

We formulate **Embodied Action Memory (EAM)** as a setting in which an embodied agent's decision context evolves with experience. Later decisions depend on a memory state formed over time, together with any observations supplied when the decision is requested. Let o_t be the observation at time t, $\mathcal{M}_t$ the memory after processing observations through t, and x_τ a later language question, goal, or decision context. EAM captures this setting as

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

where the formation function F updates memory online and the decision function G acts on the memory available at time τ. Our benchmark enforces this setting through time-restricted memory questions and post-observation planning goals.

The decision d_τ is language-level: a multiple-choice answer for memorization, or a sequence of grounded action–entity–location triples for planning. In our setting, memory is experience-dependent: decisions can use routines, preferences, action history, objects, and locations observed for that participant instead of a generic action script. EAM therefore treats F as a structured formation process, not an archival write operation. It preserves the spatial, entity-state, temporal-action, and consolidated-regularity continuities instantiated by MEMORA in §IV.

IV. Method

We present MEMORA as an implementation of the EAM formulation. §IV-A describes memory formation, where observations are encoded, edited into typed stores, and consolidated across repeated experience. §IV-B describes how the formed memory is exposed for retrieval and planning, and §IV-C defines the evaluation suite. The two central design choices are four typed stores matched to embodied continuities, and store-specific write/update rules that combine deterministic maintenance, LLM-based entity editing, and offline consolidation.

A. Memory Formation: Encoding, Editing, and Consolidation

MEMORA operationalizes EAM with four typed stores: **Environment Memory**, **Entity Memory**, **Activity Memory**, and **Inferred Knowledge** (Table I).

The stores separate continuities that change at different rates: spatial context is relatively stable, object identity persists through state change, actions provide ordered event evidence, and routines or preferences emerge across repeated experience. Motivated by multi-store views of memory [4] and distinctions between episodic traces and semantic generalisations [44, 33], MEMORA maintains $\mathcal{M}_t = (\mathcal{M}_t^{\text{env}}, \mathcal{M}_t^{\text{ent}}, \mathcal{M}_t^{\text{act}}, \mathcal{M}^{\text{inf}})$. The memory-store assignment determines the update rule for each observation: deterministic operations handle stable structure, while the Memory Editor performs semantic revision.

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 III:

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

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

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 [7, 49], the *Memory Editor* selects $\delta_j \in \Delta = \{\text{ADD}, \text{UPD}, \text{DEL}, \text{NOOP}\}$ 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 [43] over repeated entity and activity evidence. This editable state lets later observations update object histories or delete inconsistent identifications, reducing the chance that early segment-level errors persist unchanged into downstream retrieval. JSON-style records, entity-edit diagnostics, and memory-store composition are in Appendix 0a.

MEMORA therefore maintains an action-relevant memory state rather than an append-only observation log; Figure 2 quantifies a typical 18× reduction relative to the unedited observation stream on the 18-participant EPIC-KITCHENS-100 extension corpus (state histories preserved).

B. Memory-Guided Retrieval and Planning

Once memory has been formed, MEMORA is exercised through two interfaces. *Retrieval* is an evaluation interface: given a question q at time t_{ask}, the agent recovers experience-grounded evidence from the available memory snapshot $\widetilde{\mathcal{M}}_{t_{\text{ask}}}$ and selects an answer a only when the memory supports it. *Planning* is an application interface: given a goal g, the agent turns the memory state into a grounded plan $\pi = ((a_1, e_1, \ell_1), \ldots, (a_n, e_n, \ell_n))$, where each step specifies an action a_k, an entity e_k, and a location ℓ_k drawn from memory.

The two interfaces share an iterative ReAct loop [52] whose primary read tool is a *type-aware* `search`: under the hood it dispatches to four typed indices — `objects` (**Entity Memory**), `activities` (**Activity Memory**), `environment` (**Environment Memory**), and `patterns` (**Inferred Knowledge**) — with an optional `category` argument when the agent already knows which store to query. Additional memory-read tools expose the structured access patterns enabled by typed memory: `get_state_at_time` and `get_object_history` support time-restricted reasoning by rolling entities back through their `state_history` and excluding **Inferred Knowledge** records that post-date t_{ask}, so the agent cannot use evidence consolidated after the question time; `get_routine_skill` and `get_preferences` expose consolidated **Inferred Knowledge** for planning. The agent uses the same `search` tool across conditions, while MEMORA internally routes each query to the appropriate typed memory index.

At read time the four stores partition by role: **Inferred Knowledge**/**Activity Memory** supply *procedural templates* ("how to do it"), and **Entity Memory**/**Environment Memory** supply *physical grounding* ("which object, where, with what attributes") — a partition that matters most when no observed routine matches the goal (Generalize). Tool signatures, prompts, iteration budgets, and an end-to-end trace are in Appendix 0a.

C. MEMORA-Bench: An Evaluation Suite for Embodied Action Memory

Probing embodied action memory requires both *faithfulness to past experience* and *support for future action* over memory *constructed* from continuous embodied experience. This combination is absent from long-video QA [11, 32] and conversational-memory benchmarks [30, 48]. MEMORA-Bench therefore pairs two interfaces over the same experience-grounded memory state, formed from 45 hours of EPIC-KITCHENS-100 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), each item including an explicit information-not-available option to separate missing memory evidence from forced guessing. **MEMORA-Planning** probes prospective utility by producing grounded plans for experience-grounded

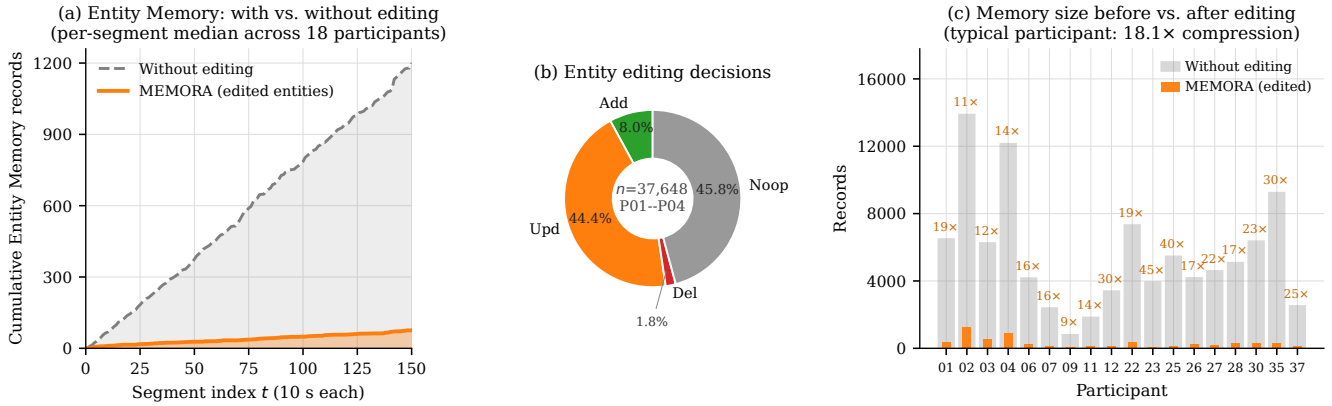

Fig. 2: **Active memory formation in Entity Memory (18 participants). (a)** Cumulative records with and without the editor (per-segment median, first 150 segments). **(b)** Editor decision mix (n=37,648, P01–P04): NOOP (45.8%) and UPD (44.4%) dominate ADD (8.0%) with rare DEL (1.8%) — the editor mostly *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 C, V0d.

requests, split into **Replay** (observed workflows) and **Generalize** (transfer, composition, or fully novel goals). The benchmark goals follow an experience-dependent view of embodied memory: they ask whether an agent can retain preferences, habits, routines, episodic action evidence, and the observed context needed for future action. We instantiate these goals from human-annotated EPIC action sequences, task templates, and verification rules; evaluated agents are defined separately by the memory interface exposed at test time, with MEMORA following an experiential consolidation design that separates episodic evidence, entity-state memory, spatial context, and consolidated routines or preferences. EAM-QA and Generalize references are annotation- or task-grounded; Replay metrics use matched participant routines, objects, and preferences when those axes are defined. Released benchmark items are static artifacts and the MEMORA pipeline itself runs entirely on open-weight models (Appendix V0k), so reproducing the headline results does not require closed-weight LLM access. Construction and quality-control details are in Appendices G, R0b, and G; dataset, conditions, and metrics follow in §V-A.

V. EXPERIMENTS

A. Experimental Setup

a) Evaluation logic.: The experiments isolate three parts of the MEMORA design. Online editing is tested by measuring whether it reduces noisy entity accumulation and improves memory use. Offline consolidation is tested by comparing full MEMORA with MEMORA-Episodic on cross-event memory and out-of-distribution planning. Typed memory is tested against flat text and entity-graph memories under the same planner backbones. The robot study then checks whether the generated language-level plans can ground two tabletop executions.

b) Data and protocols.: We evaluate on the 45-hour EPIC-KITCHENS-100 extension video subset used by MEMORA-Bench [2], covering all 18 participants for whom consolidated **Inferred Knowledge** memory banks are built (8–15 sessions each), with tasks grounded in each participant's observed action history. The subset contains 201 egocentric videos and yields 5,653 object-state entries with 1,957 unique object-name strings in the constructed memory artifacts, providing substantial variation in kitchen layouts, object instances, and recurring routines within the tabletop manipulation domain. **MEMORA-Planning** is the primary robotics-facing protocol, with 207 in-distribution **Replay** tasks and 153 out-of-distribution **Generalize** tasks; Replay is scored against matched memory-derived routine/object/preference references when defined, while Generalize uses verified task-level order references. Subsets are reported in Appendix R0b. **EAM-QA** complements planning with N=2,212 multiple-choice items (four content options A–D plus an explicit information-not-available option E) across the four EAM types defined in §IV-C, testing whether the formed memory faithfully preserves the experience-grounded evidence that planning later relies on.

c) Conditions and metrics.: We compare a controlled seven-condition ladder: **Parametric** with no memory, **Flat-1D** chronological text memory [56] and **Graph-2D** entity-relation memory [28] in both *raw* and MEMORA-edited variants, **MEMORA-Episodic** without offline consolidation, and full **MEMORA**. All matched comparisons fix perception, retrieval/tool interface, and backbone LLM; task specifications come from the benchmark-construction pipeline, while agent variants differ only in the memory interface exposed at test time. Thus raw/edited Flat-1D and Graph-2D comparisons isolate write-time memory construction, while MEMORA-Episodic versus full MEMORA isolates offline consolidation under the same downstream agent loop. EAM-QA emphasizes experience-dependent memory questions missed by Parametric, where the answer is determined by observed action history rather than generic kitchen priors; complementary evaluations on the full benchmark and memory-grounded questions are in Appendix L.

Planning keeps Parametric as a real prior baseline and reports **Robot-Grounded Plan score** (**RGP**), the mean of **OrderExec** (step order), **KeyObj** (required objects), and **PrefAdh** (matched preferences); reference provenance and auxiliary planning metrics are in Appendices M0b and S. Unless noted, perception and editing use Qwen2.5-Omni-7B and Qwen3-30B-A3B-Instruct-2507; both interfaces use the same four open-weight answer backbones, with per-role model identifiers, vLLM serving, and context budgets in Appendix X0d. Table II links the benchmark probes to planning use; Figures 3 and 4 show the main planning and robot-deployment results.

B. MEMORA-Planning Results

Planning tests whether formed memory can guide future action beyond retrospective question answering. Replay is scored against matched memory-derived routine, object, and preference references when defined; Generalize requires transfer or composition when no stored episode matches the goal and uses verified task-level order references.

a) MEMORA is strongest on Generalize tasks requiring transfer and composition.: A MEMORA condition achieves the highest RGP in each Replay and Generalize setting across the four answer backbones. The gain is consistently larger on Generalize than on Replay, indicating that structured memory is especially useful when the planner must compose remembered routines, objects, locations, and preferences for a goal not directly observed as a single prior workflow. This is the planning version of the behavior illustrated in Table II: the agent first retrieves reusable procedural evidence, then grounds it in remembered entities and places. Figure 3 pairs a real Generalize trace with the Gemma-4-31B-it case: all non-MEMORA retrieval baselines fall below the no-memory prior, while MEMORA improves RGP by $+16.4\%$. The same ranking pattern holds on Qwen3.6-35B-A3B; detailed results for each backbone and cross-backbone scaling are in Appendix M0b, Figure 15. Additional MEMORA-Planning examples generated from EPIC-KITCHENS-derived tasks are shown in Appendix 18.

b) Offline consolidation helps when planners can compose it.: Comparing MEMORA with MEMORA-Episodic isolates the contribution of offline consolidation. On Replay, MEMORA-Episodic matches or exceeds full MEMORA for several backbone settings because the relevant routine is already present as episodic evidence. On Generalize, full MEMORA helps the two strongest reasoning backbones, Qwen3.6-35B-A3B ($+.025$) and Gemma-4-31B-it ($+.022$), where reusable routines and preferences in **Inferred Knowledge** can be composed with entity and location evidence. Smaller backbones are neutral or negative under the same comparison, indicating that consolidated regularities help when the planner can combine them with episodic evidence. Thus the effect is not simply "more memory": consolidation helps most when the answer model can use it as an intermediate abstraction for a new goal. On Gemma-4-31B-it, per-axis tables attribute the largest relative gains to **OrderExec** (recovering reference steps in order; $+43\%$) and **PrefAdh** (reflecting matched preferences; $+14\%$), while **KeyObj** coverage remains competitive with the parametric prior (Appendices M0b, S).

c) Diagnostics identify where the planning gains come from.: The central mechanism is that MEMORA gives the planner separable evidence for procedure, preference, object state, and location, instead of one undifferentiated retrieved chunk. Tool logs reflect this division of labor: Flat-1D primarily issues keyword `search` queries, while MEMORA uses a typed cascade over routines, preferences, entities, and activities; Appendix M0b, Figure 16 gives the store-level breakdown. This helps explain why flat retrieval can help Replay yet hurt Generalize: it exposes evidence, but does not separate reusable procedure from physical grounding. Ablations further test whether these gains depend on memory construction. Reducing the Memory Editor from Qwen3-30B-A3B to Qwen3-14B costs 5.8 EAM-QA points and raises information-not-available selections from 42.8% to 62.3%, suggesting that weaker write-time editing makes memory less usable, not merely smaller. Replacing Omni perception with VL-only perception costs 2.7 EAM-QA points, and removing **Inferred Knowledge** shifts 35% of planning calls toward episodic activity search rather than removing the need for consolidated routines; full ablations are in Appendix V and Table XX. Additional breakdowns, alternative grounding metrics, and worked examples are in Appendix M0b.

C. MEMORA-Embodied Memory Assessment Results

a) MEMORA improves both cross-video and single-video memory probes.: Table III reports per-type accuracy under the experience-dependent, memory-grounded evaluation subset; Appendix L reports complementary evaluations on the full benchmark and memory-grounded questions. MEMORA is best on all four question types and overall, leading the strongest controlled baseline, FLAT-1D after online editing, by $+20.5$ points on Gemma-4-31B-it. The lead holds on cross-video probes SPREF, SHABIT, and SROUTINE, and on the single-video probe ERECALL, indicating that the same memory state supports aggregation across sessions and retrieval within a session. The pattern follows the benchmark design in Table II: repeated evidence benefits from consolidation into explicit regularities, while within-video recall benefits from edited object histories rather than duplicated observations.

b) Both online editing and offline consolidation contribute.: The raw and processed baseline rows in Table III separate the two effects. Online editing before graph construction lifts Graph-2D by $+7.0$ points but Flat-1D by only $+2.0$ points: chronological text cannot preserve typed state revisions and is vulnerable to noisy passages [42, 54]. *Offline consolidation*, measured as MEMORA minus MEMORA-Episodic, adds $+14.4$ points on top of the multi-store online memory. The gain is concentrated on cross-event types, with at least $+9.4$ points on each of SPREF, SHABIT, and SROUTINE; the factorial store/editing ablation and a negative result on overly elaborate consolidation are in Appendix G. The larger gains on these cross-event probes suggest that consolidation is most useful where single-episode lookup is insufficient and

TABLE II: **How MEMORA-Bench connects memory formation to planning use.** Each row states the capability being tested, shows a real MEMORA output, and explains why that capability matters for memory-grounded robot planning. Aggregate baseline comparisons are reported in Figure 3.

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.

repeated observations have to be summarized into participant-level patterns.

c) The gains are consistent across backbones.: The Table III pattern replicates on the other three open-weight backbones: MEMORA leads the strongest controlled baseline for every backbone and question type, and the offline-consolidation lift remains positive throughout. Per-backbone tables, analyses of model capacity and memory-construction effects, and complementary evaluation views are in Appendix M and Appendix L.

D. Qualitative Robot Deployment

MEMORA's language-level plans are executed on a Unitree G1 with a fixed rule-based low-level controller. The tasks require experience-grounded choices: *Prepare Drink* selects Coca-Cola and the orange cup despite distractors, while *Breakfast* selects the yellow bowl, Cinnamon Tea, yellow spoon, and bread in the remembered order. Figure 4 shows the qualitative deployment check on two prepared tabletop tasks using memory-grounded plans. Full traces are in Appendix S0d, Figures 19 and 20.

VI. Discussion

a) Memory as a substrate for robot planning.: MEMORA treats memory as part of the planning interface. The core separation is procedural versus grounding evidence: **Inferred Knowledge** and **Activity Memory** provide routines and temporal traces, while **Entity Memory** and **Environment Memory** resolve the objects, states, and places needed for the current plan. This separation becomes most visible in Generalize tasks, where no single stored episode fully specifies the future goal.

b) Reading the evidence.: The controlled comparisons make the role of the memory lifecycle visible. Online editing changes the write path, reducing duplicated entity evidence while preserving state histories. Offline consolidation changes the memory state, improving cross-event questions and Generalize planning when the answer backbone can compose reusable regularities with episodic evidence. Typed retrieval changes the read path, giving the planner distinct procedural and grounding evidence instead of one undifferentiated context.

c) Relation to robot policies.: Modern robot policies increasingly combine language goals, visual observations, and learned action priors. This includes VLA policies that map instructions and images to actions, world-action models (WAMs) that reason over action-conditioned futures, and lower-level controllers that execute motions in a particular embodiment. MEMORA addresses the memory side of this stack: long-horizon context accumulated through prior embodied experience. It records what tends to remain stable, what has changed, which procedures have been observed, and which regularities are specific to the participant. Such semantic-procedural memory can guide policy-level action selection, while VLA policies, WAMs, dynamics models, or controllers handle embodiment-specific execution. In this sense, egocentric video can support two complementary roles: learning action-capable robot policies and forming the memory those policies or planners can consult before acting.

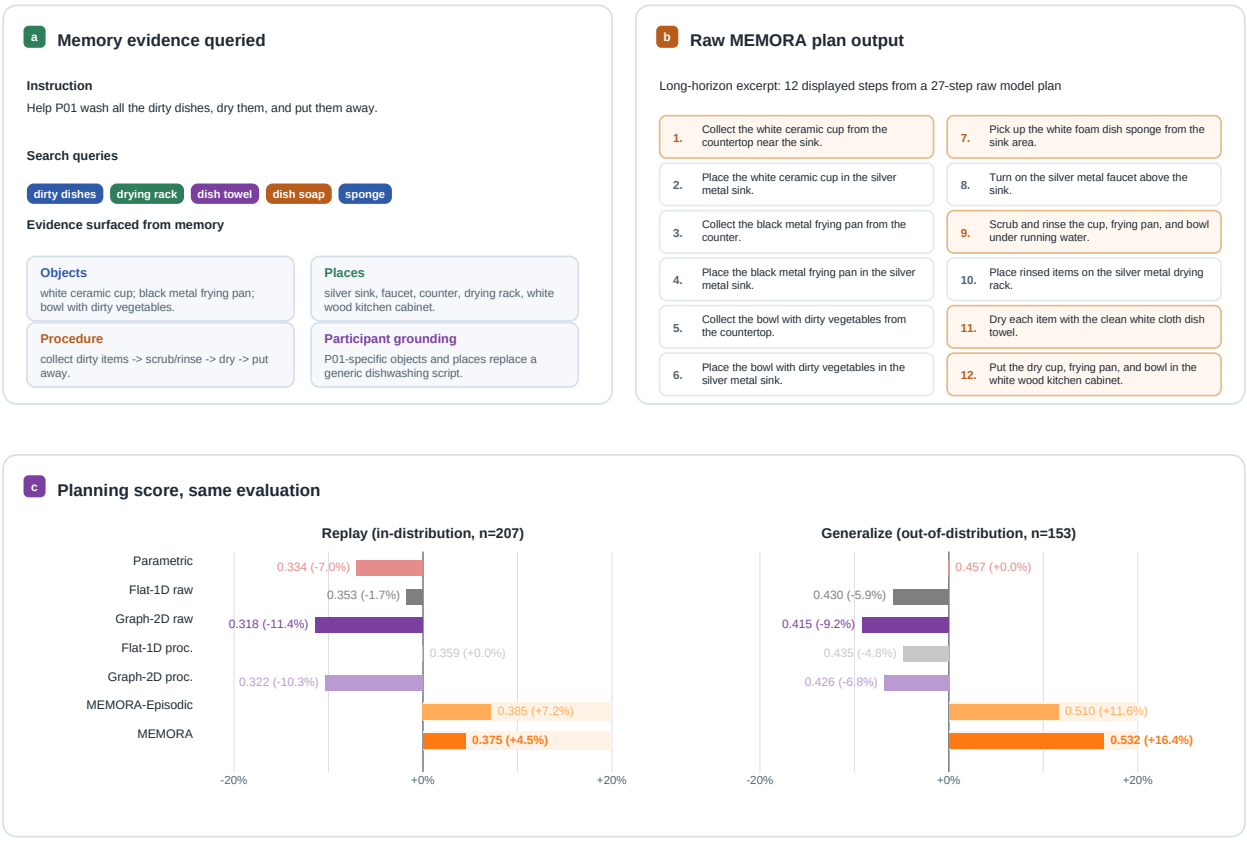

Fig. 3: **MEMORA-Planning connects remembered evidence to long-horizon plans and aggregate gains. (a)** A real Generalize task queries memory-derived evidence for dirty dishes, drying rack, dish towel, dish soap, and sponge evidence. **(b)** The raw MEMORA model response produces a 27-step plan; the figure shows a compact excerpt that collects dirty items, grounds tools and places, sequences scrub/rinse/dry operations, and puts items away. **(c)** On Gemma-4-31B-it, full MEMORA improves out-of-distribution Generalize RGP by $+16.4\%$ relative to the strongest non-MEMORA baseline. Cross-backbone and per-axis details are in Appendix M0b.

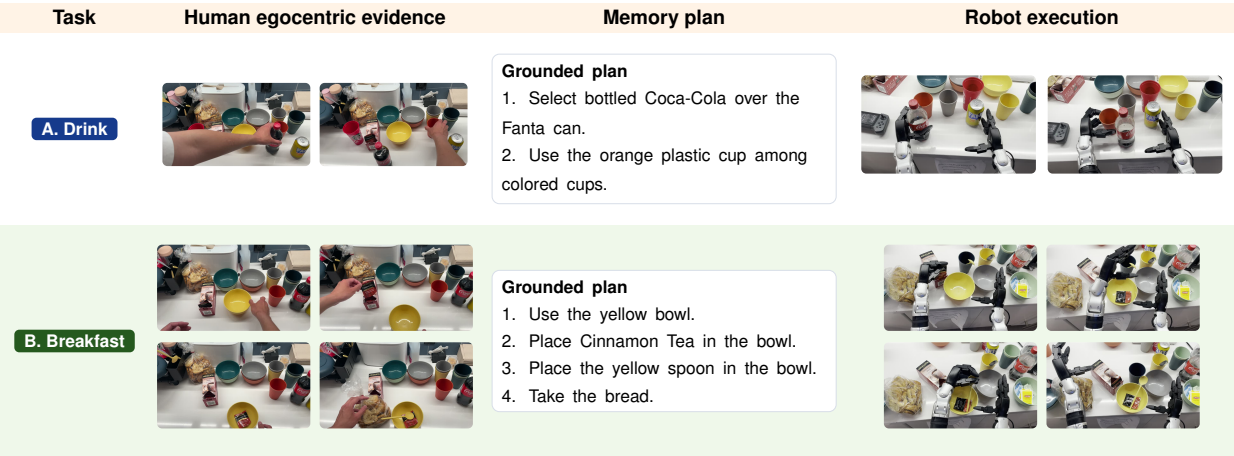

Fig. 4: **Physical robot demonstration.** MEMORA uses memory formed from human egocentric video to generate language-level plans executed by a rule-based Unitree G1 for *Prepare Drink* and *Breakfast*. Generated plans are in Appendix XVII; full per-frame traces are in Appendices 19 and 20.

TABLE III: **MEMORA-Embodied Memory Assessment per-type accuracy (%) on Gemma-4-31B-it** (N=1,122 EAM-QA questions under the experience-dependent, memory-grounded evaluation subset 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 subset focuses on questions it answered incorrectly; see Appendix L. Per-backbone reproductions are in Appendix Table IX.

	Cross-video			Single-video	
Memory condition	**SPREF**	**SHABIT**	**SROUTINE**	**ERECALL**	**Overall**
Baselines built directly from the perception stream					
Flat-1D [56], raw	63.2	36.8	53.8	51.2	52.0
Graph-2D [28], raw	50.9	21.4	37.1	55.4	40.0
Baselines rebuilt after MEMORA's online editing					
Flat-1D [56], processed	62.4	41.4	56.8	53.0	54.0
Graph-2D [28], 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**

d) Why egocentric experience is the source.: Egocentric video gives MEMORA a first-person record of how a participant actually uses a space, beyond how a generic task could be performed. This matters for planning because many goals in familiar spaces are under-specified by the instruction alone; in our tabletop examples, *prepare drink* depends on the remembered cup and beverage, and *breakfast* depends on remembered object choices and ordering. The benchmark and robot demonstration use this property in different ways. MEMORA-Bench measures whether memory formed from many sessions improves retrieval and planning at scale, while the robot demonstration illustrates the same kind of participant grounding in two physical tabletop tasks.

e) Why both benchmark protocols are needed.: Together, EAM-QA and MEMORA-Planning test whether formed memory is both faithful to past experience and useful for future goals: EAM-QA probes remembered evidence, while planning stresses prospective composition across remembered routines, entities, and locations. Within planning, Replay checks recovery of observed workflows, while Generalize asks whether typed memory exposes reusable structure for new goals.

VII. CONCLUSION

We introduce **MEMORA**, a typed memory framework that turns egocentric experience into editable, consolidated memory for planning, and **MEMORA-Bench**, a 45-hour evaluation suite for memory faithfulness and memory-grounded planning. Across four open-weight backbones, MEMORA outperforms flat-text and entity-graph memories, with the largest planning gain on out-of-distribution GENERALIZE tasks: up to $+16.6\%$ relative RGP on Qwen3.6-35B-A3B. A qualitative two-task robot deployment check uses the same memory to generate remembered-context language plans for downstream execution. Together, the results show that how embodied experience is formed, edited, consolidated, and queried matters when future goals require more than replaying a matching episode. This memory layer can complement egocentric imitation learning and vision–language–action systems: those methods learn executable behavior from human video, while MEMORA organizes the same kind of experience into experience-grounded context, routine cues, and procedure choices for planning.

LIMITATIONS

a) Domain and horizon.: The current benchmark is built from kitchen activity in the EPIC-KITCHENS-100 extension videos [2], with at most 15 sessions per participant. This is long enough to expose recurring places, objects, and routines, but not enough to study months-long drift, seasonal reorganization, or rare events. Extending EAM beyond this kitchen-centered benchmark may require policies for capacity, forgetting, and deciding when repeated observations revise rather than reinforce a regularity.

b) Pipeline dependence.: MEMORA depends on the quality of perception and write-time editing. The ablations in Table XX show that weaker perception models or weaker memory editors reduce memory-assessment performance. Because memory is formed from perceived segments, missed objects cannot be recovered unless they appear in later evidence, and repeated perception errors may enter consolidated knowledge if they are not contradicted by subsequent observations.

c) Robot deployment and privacy.: The Unitree G1 experiment studies language-level memory grounding in a prepared tabletop scene with a fixed controller. Future work can connect MEMORA to full closed-loop robot autonomy, including navigation, deformable-object manipulation, online perception updates during execution, and recovery from failed grasps. Because persistent embodied memory is participant-specific, practical deployment also requires consent, inspection and deletion tools, access control, and clear limits on memory sharing.

REFERENCES

[1] 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, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

[2] Dima Aldamen, Davide Moltisanti, Evangelos Kazakos, Hazel Doughty, Jonathan Munro, William Price, Michael Wray, Tobias Perrett, and Jian Ma. Epic-kitchens-100, 2020. URL https://data.bris.ac.uk/data/dataset/2g1n6qdydwa9u22shpxqzp0t8m/.

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

[4] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. volume 2 of *The Psychology of learning and motivation*, pages 89–195. Academic Press, New York, 1968.

[5] 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[6] 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, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[7] Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

[8] Daesol Cho, Youngseok Jang, Danfei Xu, and Sehoon Ha. Egoavflow: Robot policy learning with active vision from human egocentric videos via 3d flow, 2026. URL https://arxiv.org/abs/2602.22461.

[9] Jeremy R. Cole, Michael J. Q. Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions, 2023. URL https://arxiv.org/abs/2305.14613.

[10] Pengfei Du. Memory for autonomous llm agents: Mechanisms, evaluation, and emerging frontiers. *arXiv preprint arXiv:2603.07670*, 2026.

[11] 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, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

[12] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2022. URL https://arxiv.org/abs/2110.07058.

[13] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[14] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat,

and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training, 2020. URL https://arxiv.org/abs/2002.08909.

[15] Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video, 2025. URL https://arxiv.org/abs/2505.11709.

[16] 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. Inner monologue: Embodied reasoning through planning with language models, 2022. URL https://arxiv.org/abs/2207.05608.

[17] Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift, 2020. URL https://arxiv.org/abs/2006.09462.

[18] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

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

[20] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering, 2019. URL https://arxiv.org/abs/1906.00300.

[21] 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. Robomemory: A brain-inspired multi-memory agentic framework for interactive environmental learning in physical embodied systems, 2026. URL https://arxiv.org/abs/2508.01415.

[22] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing, 2025. URL https://arxiv.org/abs/2508.09976.

[23] 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. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

[24] 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. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024. URL https://arxiv.org/abs/2311.17005.

[25] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL https://arxiv.org/abs/2209.07753.

[26] Xingyao Lin, Guojin Zhong, Tianyi Lu, Ziyi Ye, Yichen Zhu, Zuxuan Wu, and Yu-Gang Jiang. Activemimic: Egocentric video pretraining with active perception, 2026. URL https://arxiv.org/abs/2606.06194.

[27] Yangcen Liu, Shuo Cheng, Xinchen Yin, Woo Chul Shin, Alfred Cueva, Yiran Yang, Zhenyang Chen, Chuye Zhang, and Danfei Xu. Egoengine: From egocentric human videos to high-fidelity dexterous robot demonstrations, 2026. URL https://arxiv.org/abs/2606.12604.

[28] Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory, 2025. URL https://arxiv.org/abs/2508.09736.

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

[30] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents, 2024. URL https://arxiv.org/abs/2402.17753.

[31] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023. URL https://arxiv.org/abs/2212.10511.

[32] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding, 2023. URL https://arxiv.org/abs/2308.09126.

[33] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. 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, 1995.

[34] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

[35] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022. URL https://arxiv.org/abs/2206.06520.

[36] Pearl Mody, Mihir Panchal, Rishit Kar, Kiran Bhowmick, and Ruhina Karani. Cranimem: Cranial inspired gated and bounded memory for agentic systems. *arXiv preprint arXiv:2603.15642*, 2026.

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

[38] Yichen Niu, Haoran Lv, Xinrui Zhang, Xueyao Wan, Shiyu Gao, Ying Ai, Hui Xu, Yongqi Hu, Hengyi Zhang, Yang Xie, Zhaxizhuoma, Yue Zhao, Zhenshan Bing, Yan Ding, and Jianxing Liu. Egoaero: Learning dexterous manipulation from a single egocentric video without object assets, 2026. URL https://arxiv.org/abs/2606.08057.

[39] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez.

Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

[40] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

[41] 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, Renee Zbizika, Jeff Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Chenyu Yang, Xi Wang, James Fort, Richard Newcombe, Josh Gao, Jason Chong, Garrett Matsuda, Aseem Doriwala, Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, and Danfei Xu. Egoverse: An egocentric human dataset for robot learning from around the world, 2026. URL https://arxiv.org/abs/2604.07607.

[42] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023. URL https://arxiv.org/abs/2302.00093.

[43] Robert Stickgold and Matthew P. Walker. Sleep-dependent memory consolidation. *Nature*, 437:1272–1278, 2005.

[44] Endel Tulving. Episodic and semantic memory. *Organization of Memory*, pages 381–403, 1972.

[45] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL https://arxiv.org/abs/2212.03533.

[46] Zhi Wang, Botao He, Kelin Yu, Seungjae Lee, Ruohan Gao, Furong Huang, and Yiannis Aloimonos. Humanego: Zero-shot robot learning from minutes of human egocentric videos, 2026. URL https://arxiv.org/abs/2605.24934.

[47] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks, 2015. URL https://arxiv.org/abs/1410.3916.

[48] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2025. URL https://arxiv.org/abs/2410.10813.

[49] 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. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning, 2026. URL https://arxiv.org/abs/2508.19828.

[50] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. Egovla: Learning vision-language-action models from egocentric human videos, 2025. URL https://arxiv.org/abs/2507.12440.

[51] Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning, 2025. URL https://arxiv.org/abs/2411.17735.

[52] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

[53] Chongrui Ye, Yuxiang Liu, Yu Wang, Haofei Yu, Yining Zhao, Ge Liu, Julian McAuley, and Jiaxuan You. Autodreamer: Learning offline memory consolidation for language agents. *arXiv preprint arXiv:2605.20616*, 2026.

[54] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context, 2024. URL https://arxiv.org/abs/2310.01558.

[55] Justin Yu, Yide Shentu, Di Wu, Pieter Abbeel, Ken Goldberg, and Philipp Wu. Egomi: Learning active vision and whole-body manipulation from egocentric human demonstrations, 2025. URL https://arxiv.org/abs/2511.00153.

[56] 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. Socratic models: Composing zero-shot multimodal reasoning with language, 2022. URL https://arxiv.org/abs/2204.00598.

[57] Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents, 2024. URL https://arxiv.org/abs/2404.13501.

[58] 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. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710.

[59] Xinyi Zheng, Yunze Liu, Chi-Hao Wu, Fan Zhang, Hao Zheng, Wenqi Zhou, Walterio W. Mayol-Cuevas, and Junxiao Shen. Spatialmem: Metric-aligned long-horizon video memory for language grounding and qa, 2026. URL https://arxiv.org/abs/2601.14895.

[60] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

APPENDIX

a) Appendix roadmap.: The appendix is organized around construction, evaluation, and controlled analysis. Appendix 0a specifies the memory-bank construction pipeline from egocentric video. Appendix G defines the retrospective Embodied Memory assessment and its temporal-control protocol. Appendix M0b specifies the MEMORA-Planning benchmark and rule-based evaluation panel. Appendix S0d documents the physical robot demonstration used as a qualitative deployment check. Appendix U0a defines controlled ablations over processor and answer-agent choices. Appendix X0d consolidates the model registry, inference stack, perception preprocessing, and retrieval/agent-loop settings shared across all of the above.

Appendix 0a specifies how MEMORA constructs the typed memory bank used by the Embodied Memory and Planning evaluations. The pipeline maps EPIC-KITCHENS-100 extension videos into the four stores defined in Section IV: **Environment Memory**, **Entity Memory**, **Activity Memory**, and offline **Inferred Knowledge**. The presentation follows processor roles rather than engineering stage names.

*b) 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 I). 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. Implementation Overview

Each video is divided into non-overlapping 10-second turns. This appendix uses implementation names for the two main method components: the *Perception Processor* implements the Segment Encoder ϕ, and the *Online Memory Processor* implements the write-time EDIT update. 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:

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

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.

B. 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 X0d) 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 preferences, habits, workflow patterns, and action sequences). These summaries are useful for preference, habit, and routine reasoning, but they are withheld for retrospective QA whenever they would expose evidence after the query anchor. 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 XXI to produce evidence-grounded generated preferences; a closed-weight alternative is reported as a sanity check in Appendix V0k.

C. Editor Activity and Memory Compression

This subsection documents the data behind Figure 2 in Method §IV-A. 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 XXI): 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

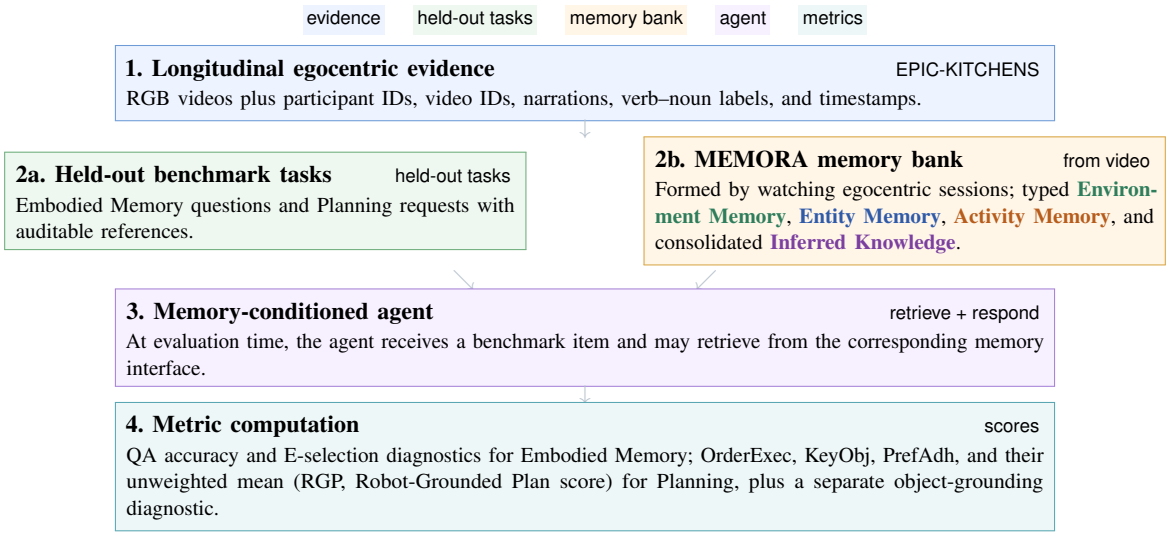

Fig. 5: **Appendix visual roadmap for MEMORA-Bench.** The appendix separates four concepts that are easy to conflate: longitudinal evidence, held-out benchmark tasks, memory-bank construction from video, and memory-conditioned evaluation. Subsequent figures expand the memory-bank construction pipeline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

TABLE IV: Main-paper memory-bank construction processors. The table specifies processor contracts (input scope, outputs, and construction notes); model identifiers appear in Table XXI and temporal preprocessing in Appendix X0d.

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 V0k.

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.

a) 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.

b) 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 (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 $18.1\times$ 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.

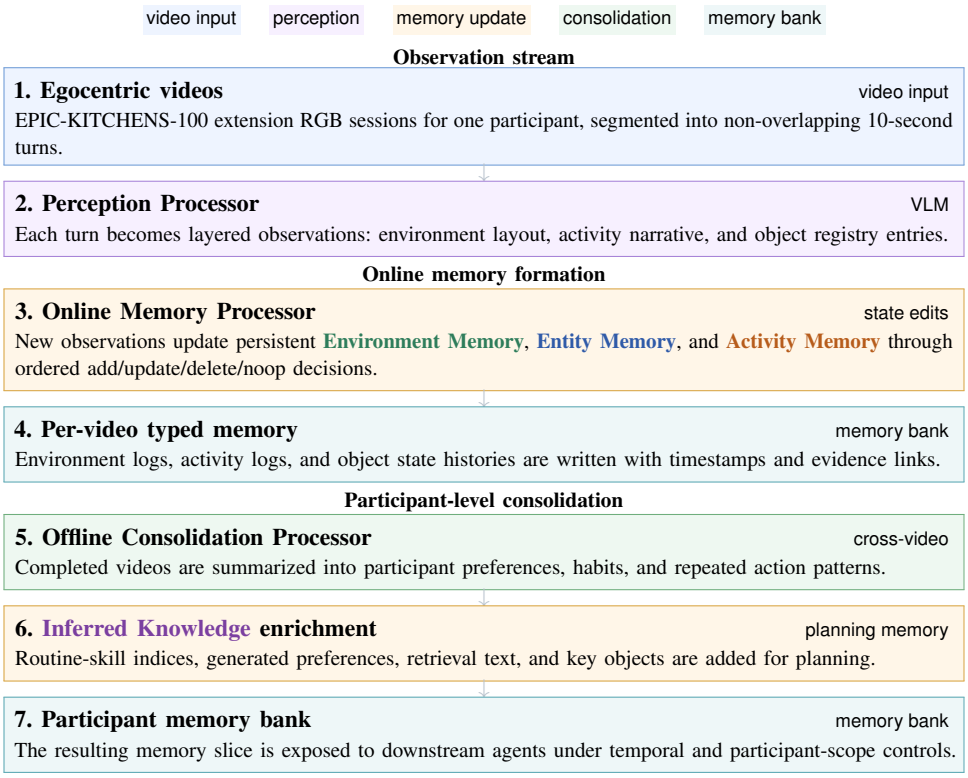

Fig. 6: **MEMORA memory-bank construction pipeline.** The system watches egocentric videos turn by turn, converts each clip into structured observations, updates typed memory online, and then consolidates participant-level routines and preferences for downstream Embodied Memory and Planning evaluations.

D. 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.

E. 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.

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.
```

Fig. 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]
```

Fig. 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.

F. Memory Stores and Leakage Control

a) Leakage control.: In retrospective Embodied Memory, *leakage control* means the agent may retrieve only evidence available strictly before the question anchor t_{ask}: activity and environment entries are time-filtered, entity states are rolled back through `state_history`, and offline **Inferred Knowledge** summaries are withheld whenever they would expose post-anchor observations. 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 V summarizes these controls for each memory store.

G. 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. Offline consolidation improves participant-level summaries but must be disabled or sliced for retrospective questions to prevent temporal leakage. 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 V: Primary memory stores and leakage controls. *Leakage control* specifies how each store is time-sliced or scope-restricted so evaluation cannot access future or out-of-scope evidence.

Store	Producer	Content	Leakage control
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	Retrieved only before the query anchor for retrospective QA
`object_registry`	Online Memory Processor	Edited entity memory with add/update/delete/noop decisions and state history	Entity states are reconstructed or rolled back for time-restricted queries
Consolidation summaries	Offline Consolidation Processor	Per-video storage preferences, organizational habits, workflow patterns, and action sequences	Withheld when consolidation would expose post-anchor evidence
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

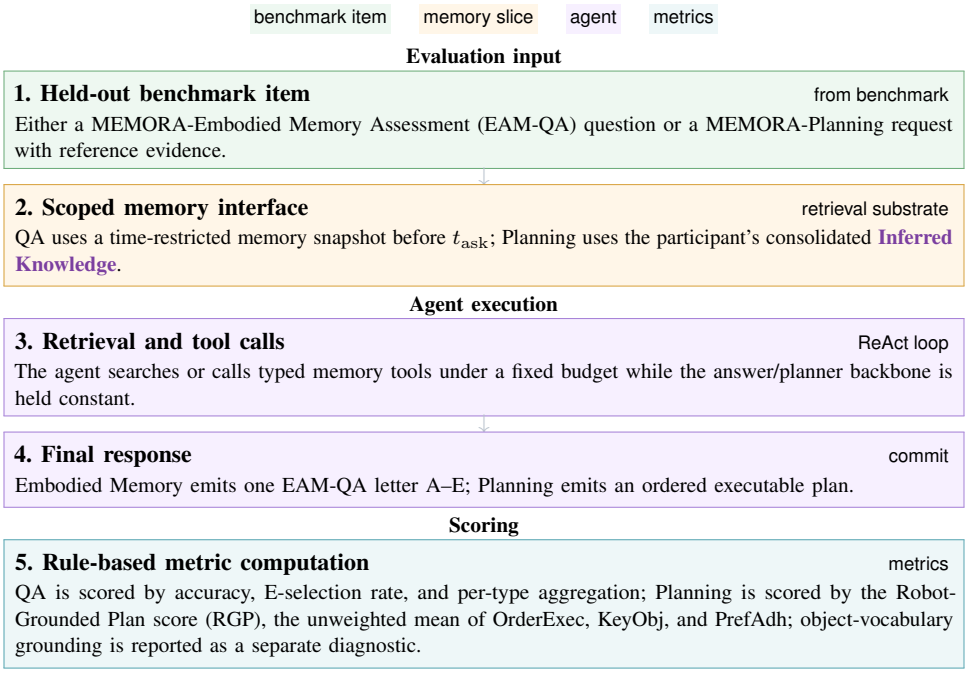

Fig. 9: **Memory-conditioned evaluation and scoring loop.** The benchmark item and the scoped memory interface are kept separate: benchmark tasks define what should be answered, while memory substrates define what the agent can retrieve before committing to an answer or plan.

Appendix G defines the retrospective QA arm of MEMORA-Bench. The **MEMORA-Embodied Memory Assessment** measures whether experience-derived memory is faithful, accessible, and temporally controlled. The annotation-grounded release supports the headline four-way split (ERECALL, SPREF, SHABIT, SROUTINE) on 45 hours of EPIC-KITCHENS-100 [2] extension video across 18 participants. Generation is architecture-agnostic: questions are mined from action annotations and do not presuppose MEMORA's four-substrate layout. §H specifies benchmark construction and question grounding; §K specifies the evaluation protocol. The EAM-QA setup evaluates retrieval from the constructed memory bank rather than generic video question answering.

H. Benchmark Construction

a) EPIC-KITCHENS inputs.: The construction builds on **EPIC-KITCHENS-100** [2], 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 further restricted to an **EPIC-KITCHENS-100 extension video whitelist** (45 hours across 18 participants), so every item remains within the same longitudinal setting.

I. Question Types and Grounding

a) Design.: Each item is *experience-dependent*: it concerns one person's observed kitchen behavior across longitudinal sessions in the same home, not generic cooking knowledge, so the benchmark tests memory formed from lived action history 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 verification) 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 verification (semantic deduplication, per-item verification, and answer re-derivation).

b) Scope of this appendix.: This appendix specifies the annotation-grounded pipeline that produces the four reported EAM-QA types in Table VI. 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 curate 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.

c) 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.

TABLE VI: 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→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 → short activity label; plausible wrong activities	One episodic item per video; answer tied to that `video_id` only

(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.

d) Per-type inputs, flow, and prompts.: Figure 10 and Table VI 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.

J. Quality Control and Diagnostics

a) 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.

b) Automated verification before release.: Every candidate passes three verification stages, each targeting a different failure mode: **(1) Semantic dedup** (embedder in Table VII; 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 verifier** (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 verification (27 removed by the quality judge alone). Representative numeric thresholds are listed in Table VII.

c) 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.

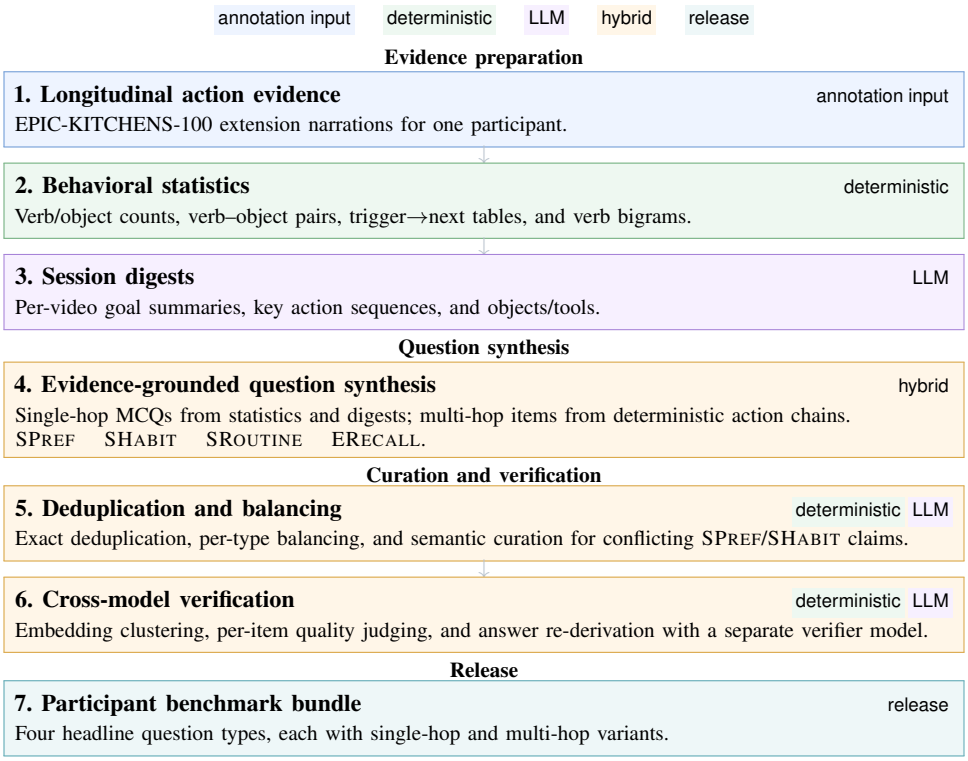

Fig. 10: **MEMORA-Embodied Memory benchmark construction pipeline.** MEMORA-Bench converts longitudinal action annotations into experience-dependent Embodied Memory questions through four phases: evidence preparation, question synthesis, curation and verification, and release. Colors indicate whether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

TABLE VII: 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` [45]	—	—	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 check	`gpt-4o-mini`	0.0	16	Verification stage
Verifier: answer re-derivation	`gpt-4o-mini`	0.0	32	Verification stage

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

d) Reproducibility.: We generate one bundle per participant on the EPIC-KITCHENS-100 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. The EAM-QA release is automatically generated and verified rather than separately human-validated. Its reproducibility comes from fixed prompts, schemas, thresholds, and annotation-grounded evidence requirements; its quality control comes from cross-model verification, answer re-derivation, and released audit metadata. This design supports scalable extension to new annotated corpora, while manual human validation remains complementary future quality control.

K. Evaluation Protocol

a) Task and metrics.: At query time (q, t_{ask}) the agent must choose one of five options $\{\text{A}, \ldots, \text{E}\}$. Options A–D are contentful answers; **E** denotes "information not available," so a

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.
```

Fig. 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.

model must recognize when the memory state lacks sufficient experience-grounded evidence 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 carries an `ask_turn_id` (or equivalent temporal anchor) so retrieval uses only observations strictly before t_{ask}; this matches embodied deployment where the agent cannot read the future and isolates memory quality from leakage. See the main method for the time-restricted snapshot $\widetilde{\mathcal{M}}_{t_{\text{ask}}}$.

b) Controlled comparison.: All memory conditions share the same perception front-end, retrieval stack, and answer-agent backbone within a column (Appendix X0d), so differences are not confounded by vision or embedding capacity. The *only* deliberate variable is the memory substrate exposed to a ReAct agent [52]: Parametric (no external memory), **Flat-1D** [56] (chronological text memory exposed via a flat `search` tool), **Graph-2D** [28] (entity-relation graph memory), or **MEMORA** (typed substrates with consolidation variants). Accuracy gaps are therefore attributed to how experience is *organized and retrieved*, not to unequal perception or prompting budgets.

c) Agent loop.: The answering agent follows the shared Embodied Memory ReAct budget in Appendix X0d (up to five `search` calls, then one letter). For Embodied Memory evaluation, the time-restricted snapshot $\widetilde{\mathcal{M}}_{t_{\text{ask}}}$ is applied: entity states roll back to t_{ask} and offline consolidation summaries are withheld so answers must come from observations strictly preceding the question anchor, not sleep-time rules that would trivialize SPREF/SHABIT items. Tool schemas and per-condition system prompts are fixed within each block; Graph2D receives schema-equivalent per-type prompts so routing verbosity does not confound substrate comparisons.

d) 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. The rate of E selections is tracked separately because flat retrieval can overproduce information-not-available answers on some commercial backbones.

e) Illustrative items.: Figure 12 provides one released record per headline type from participant P06. Fields `reasoning` and full `evidence` chains are audit metadata used during construction and quality verification; the answering agent receives only the question, four content options, and (at evaluation time) option E appended to the choice list.

L. Experience-Dependent Memory Evaluation Views

The Embodied Memory benchmark mixes *answerable* questions (ground truth $\in \{\text{A}, \text{B}, \text{C}, \text{D}\}$) with *unanswerable* probes (ground truth E), the latter required so a model cannot inflate accuracy by guessing on underspecified items. Our headline view focuses on experience-dependent memory: questions missed by the parametric (no-memory) baseline, where the answer should be determined by observed action history rather than generic kitchen priors. Within that view, we report memory-grounded answerability, the operating point where MEMORA can ground a contentful answer in retrieved memory evidence (selects one of A–D rather than E). All compared conditions are evaluated on the *same* question subset so the comparison is matched. This follows two memory-evaluation principles:

(i) the experience-dependent view isolates questions whose answer should come from the observed history of a particular

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
}
```

Fig. 12: Released Embodied Memory items (P06). Left column: SPREF and SROUTINE; right column: SHABIT and ERECALL. Option E (information not available) is appended at evaluation; it is omitted from the stored release JSON.

person, rather than from a closed-book prior or generic kitchen knowledge [20, 14, 31, 34, 35];

(ii) memory-grounded answerability asks whether the formed memory contains enough retrievable experiential evidence to support a concrete answer, rather than rewarding guesses when the memory state is incomplete [17, 9].

This is a memory-centric diagnostic: it separates two failure modes that are otherwise conflated, namely whether the memory state surfaces relevant evidence at all, and whether the answer agent can discriminate the correct option once such evidence is available. Table VIII reports MEMORA's overall accuracy and lead over the strongest controlled baseline under three views: the full benchmark, memory-grounded answerability, and the experience-dependent answerability headline view. The Parametric (no-memory) baseline is 0% in the experience-dependent view by construction and is omitted from the per-backbone tables; its full-benchmark accuracy is $\sim 20\%$ on all four backbones (random guess on 5-option MC).

a) Memory-grounded answerability rate and precision.: To check that absolute-accuracy differences across backbones reflect answer-agent capacity rather than pipeline failure, we decompose MEMORA's accuracy within the experience-dependent view into a *memory-grounded answerability rate* $\mathrm{P}(\text{MEMORA picks A–D} \mid \text{parametric wrong})$ and a *memory-grounded precision* $\mathrm{P}(\text{correct} \mid \text{A–D}, \text{parametric wrong})$; the latter equals the headline experience-dependent, memory-grounded accuracy by construction. Memory-grounded precision increases monotonically with backbone capacity — Gemma-4-26B-A4B 54.1%, Qwen3.6-27B 69.1%, Gemma-4-31B 74.5% — whereas the answerability rate varies non-monotonically (Gemma-4-26B-A4B aggressive at 67.5%, Qwen3.6-27B also high at 70.3%, Gemma-4-31B conservative at 50.7%) and stays $\geq 50\%$ everywhere, indicating that every backbone engages with the retrieved memory; what

differs across backbones is how well they discriminate among grounded options once they answer. 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 sufficient evidence to answer, not a failure of the memory pipeline that feeds it.

M. 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 III 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 experience-dependent, memory-grounded EAM-QA view of Appendix L (Table IX; the per-backbone Δ_{consol} rows summarise the per-type lift).

a) Cross-backbone consistency of the consolidation lift.: Under the unified experience-dependent, memory-grounded EAM-QA view (Table IX, 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-4-26B-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 has enough evidence for a contentful answer.

b) 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 EPIC-KITCHENS-100 extension subset, 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;
- 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-curate 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 VIII: **EAM-QA evaluation views: MEMORA overall accuracy (%) and gap to strongest controlled baseline** (SOCRATIC/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 memory-grounded answerability view reports cases where MEMORA selects a content option A–D; the headline view additionally focuses on experience-dependent questions missed by the parametric prior.

	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
Memory-grounded	56.5	-0.3	45.8	-7.2	61.3	$+5.9$	67.7	$+9.6$
Experience-dependent + memory-grounded	**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 ($n_{\text{full}}/n_{\text{evidence}}/n_{\text{headline}}$): 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 {SOCRATIC 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 IX (MEMORA 63.5%, $+11.8$ points over best controlled baseline) within ± 0.8 points.

TABLE IX: **MEMORA-Embodied Memory Assessment accuracy (%): per-backbone, per-type panel**. All four backbones report under the experience-dependent, memory-grounded EAM-QA view (Appendix L): **Gemma-4-31B-it** on the full 18-PID panel (N=1,122; mirrors Table III); **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 experience-dependent view retains only questions it answers incorrectly (0% by construction); full-benchmark Parametric accuracy and memory-grounded answerability are reported in Table VIII.

		Cross-video (semantic)			Single-video	
Backbone	**Memory condition**	**SPREF**	**SHABIT**	**SROUTINE**	**ERECALL**	**Overall**
Qwen3.6-35B-A3B[♭] *(MoE, 64K ctx)* *(N=1,141)*	Socratic (w/o memory proc.)	51.2	33.5	53.2	12.5	41.6
	Socratic (w/ memory proc.)	53.1	34.2	52.9	5.3	41.3
	Graph2D (w/o memory proc.)	50.6	27.4	54.7	48.7	45.0
	Graph2D (w/ memory proc.)	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)*	Socratic (w/o memory proc.)	48.9	41.6	49.6	65.7	49.6
	Socratic (w/ memory proc.)	49.8	38.7	49.4	61.0	48.5
	Graph2D (w/o memory proc.)	51.1	29.6	53.3	53.5	46.8
	Graph2D (w/ memory proc.)	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)*	Socratic (w/o memory proc.)	51.0	28.0	42.0	59.4	42.4
	Socratic (w/ memory proc.)	53.9	31.3	42.2	62.4	44.5
	Graph2D (w/o memory proc.)	40.4	16.3	31.6	60.2	32.5
	Graph2D (w/ memory proc.)	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)*	Socratic (w/o memory proc.)	63.2	36.8	53.8	51.2	52.0
	Socratic (w/ memory proc.)	62.4	41.4	56.8	53.0	54.0
	Graph2D (w/o memory proc.)	50.9	21.4	37.1	55.4	40.0
	Graph2D (w/ memory proc.)	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.
[♭] Experience-dependent, memory-grounded view (Appendix L): Gemma-4-31B-it on the full 18-PID EAM-QA panel (N=1,122; mirrors Table III); 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 on which all seven memory conditions are currently materialized; the full 18-PID Qwen3.6-35B-A3B panel will be released alongside the code and benchmark, and the 8-PID evaluation-view panel reported in the [♯] footnote of Table VIII indicates 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.

Appendix M0b 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 [2]: the reference sequence is an observed segment of narrated actions, but the planner is evaluated on whether it can produce an executable, experience-grounded 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.

N. Task Construction

a) 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.

b) Design principles.: Planning items are *experience-dependent*: 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.

c) 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 $i-1 \rightarrow i$ 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, segments with more than 15 actions are split into near-even chunks, and the fixed task set is shared by all memory conditions. The resulting candidates are classified into the five task types in Table X. Filler pick-and-place verbs are removed before classification so that superficial object motion does not dominate the task label.

d) 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.

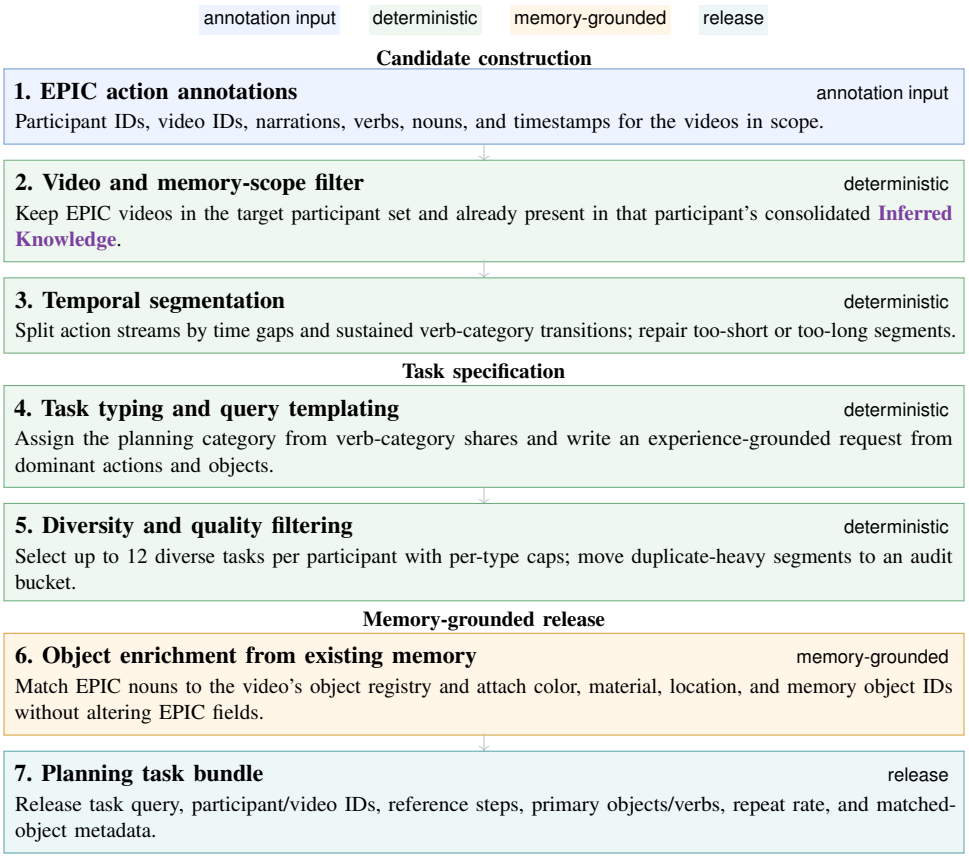

Fig. 13: **MEMORA-Planning benchmark construction pipeline.** The flow converts EPIC action annotations into prospective, experience-grounded planning requests. Unlike the memory-bank construction flow in Figure 6, this figure describes held-out benchmark construction; the memory bank is only read during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

TABLE X: 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.

TABLE XI: 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 X
`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

*e) 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.

f) 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.

O. Memory Conditions and Tools

a) Controlled comparison.: The evaluation compares three memory conditions while holding the planning model, benchmark tasks, and shared runtime stack (Appendix X0d) fixed. This isolates the effect of memory organization and tool interface. The linear-memory baseline follows the Socratic-models style of exposing a text memory through language-model interaction [56]. Table XII summarizes the conditions.

TABLE XII: Head-to-head planning conditions. MEMORA exposes consolidated **Inferred Knowledge** through routine/preference tools, **Entity Memory** tools for objects/states, and **Activity Memory** tools for activity evidence.

Condition	Memory	Tools	Prompt behavior
`no_memory`	none	none	Single-shot planning from the request only.
`socratic`	linear Socratic memory	`search`	Search-only access to a text memory of observations.
`memora`	consolidated **Inferred Knowledge**	full toolbox	Routine-first prompt prioritizing `get_routine_skill`, preferences, and **Entity Memory** grounding.

b) Planner and runtime.: Planning-specific runtime settings beyond the shared stack in Appendix X0d 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.

c) ReAct loop and plan parser.: Memory-backed conditions run a ReAct-style loop [52] under the shared agent-loop budget in Appendix X0d (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.

d) 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.

e) Planning prompt templates.: Figure 14 summarizes the prompt templates used by the planning conditions and forced-answer mode. The no-memory, Socratic, and MEMORA conditions differ in available memory and tools.

No memory

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

Socratic 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.
```

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. ...
```

Fig. 14: **Planning prompt templates.** The panel compresses the four prompt surfaces into their operational differences: available memory, tool surface, retrieval priority, and final-answer constraints.

P. Metrics

a) 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 XIII: 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	Defined per-task axes above	All component axes are undefined

b) 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 chosen 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:

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

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.

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

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:

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

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 per-task mean of three robot-followability axes:

$$\mathrm{RGP}(P) = \mathrm{mean}\big(\mathrm{OrderExec}, \mathrm{KeyObj}, \mathrm{PrefAdh}\big). \quad (7)$$

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 §S 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.

Q. Results and Diagnostics

a) Cross-backbone summary.: Figure 15 summarises 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 §V-B. First, MEMORA's gain is at least as large on Generalize as on Replay for three of the four backbones, indicating that structured, consolidated memory matters most when no verbatim segment is available to retrieve. 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.

b) 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 §V-B is available in the figure companion panel.

c) Per-condition numerical results: Qwen3.6-35B-A3B.: Table XIV 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.

d) Per-store tool-use distribution.: Figure 16 aggregates every logged `tool_call` across all 1,440 planning tasks (4 backbones $\times$ 2 suites $\times$ 18 PIDs $\times$ 12 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), so any architecture that exposes only one store would systematically disadvantage one backbone family. 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**, suggesting that consolidated routines absorb queries that episodic activity search alone answers noisily. 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.

e) Complementary evidence from EAM-QA.: The planning-side distribution above is partial because the planning prompt does not advertise an environment-search tool; spatial information rides on **Entity Memory** object returns. On EAM-QA, where the agent is given a single unified `search` entry point with optional `category`$\in$ {`objects`,`activities`,`environment`,`patterns`}, the same backbone uses that one tool for 95.9% of all calls (Figure 17, aggregated from 7,068 questions and 17,220 tool calls across four backbones and four EAM-QA question types: ELoc, EOrder, SPref, ERecall). That call pattern is consistent with **Environment Memory** being read on essentially every query: the unified search routes internally through all four typed indices and merges the results, so **Environment Memory** contributions surface inside an `environment`-category result list rather than as a separate tool invocation. The two views together show that all four stores are accessed in practice — planners reach them through typed entry points the prompt advertises, while EAM-QA agents reach them through a single dispatching call.

f) Illustrative trace.: Figure 18 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.

```
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.
```

Fig. 18: **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.

R. Generation pipeline reproducibility

a) 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.

b) 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. The second family, *composition*, samples a high-level goal that is *not* present in `routine_skills`

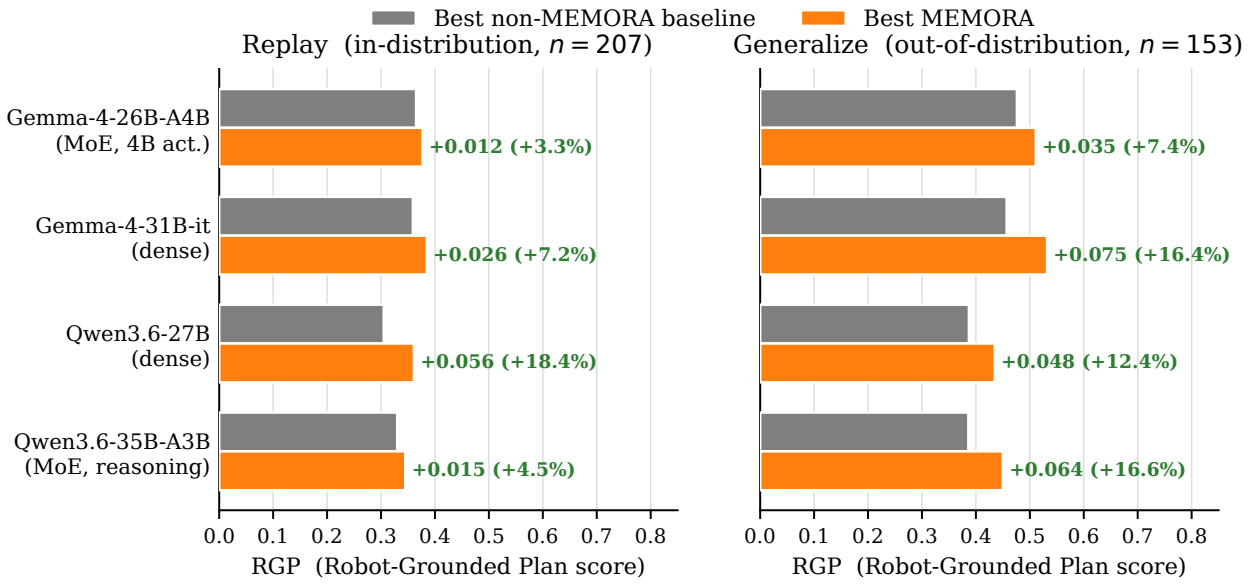

Fig. 15: **Cross-backbone RGP summary** (companion to Figure 3). Grey = best non-MEMORA baseline; orange = best MEMORA variant; annotations give Δ RGP and relative gain.

TABLE XIV: 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**

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 penalises 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.

S. *Object-grounding diagnostic*

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

a) 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

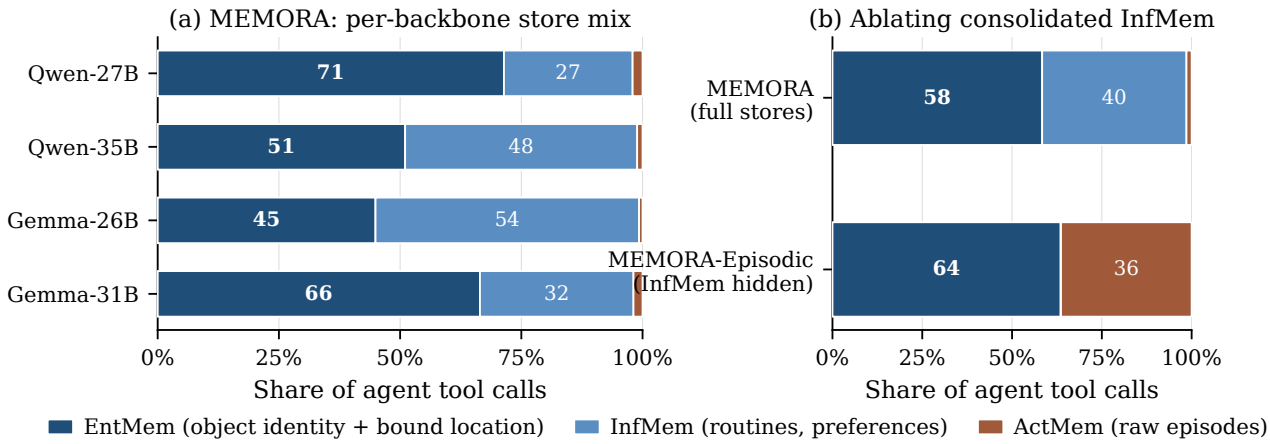

Fig. 16: **Per-store contribution of agent tool calls during planning** (1,440 tasks = 4 backbones × 2 suites × 18 PIDs × 12 tasks). **(a)** Under the main MEMORA configuration, every backbone draws on multiple typed stores, but with markedly different mixtures: Gemma4-26B-A4B-it routes 54% of its tool budget to **Inferred Knowledge** (routines, preferences), while Qwen3.6-27B routes 71% to **Entity Memory** (object identity). A single-store substrate would penalise at least one backbone. **(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, direct evidence that consolidated routines are doing work that episodic activity search alone cannot supplant. **Environment Memory** is omitted from the figure because spatial information is bound to object records in the planning prompt (every `search_objects` return ships `location` attributes), so a separate environment-search tool is not advertised; the EAM-QA agent instead reaches **Environment Memory** through the unified `search` interface (Figure 17).

TABLE XV: 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

participant's KB-vocabulary noun set (§M0b). 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?"

b) Results across all completed cells.: Table XVI 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.

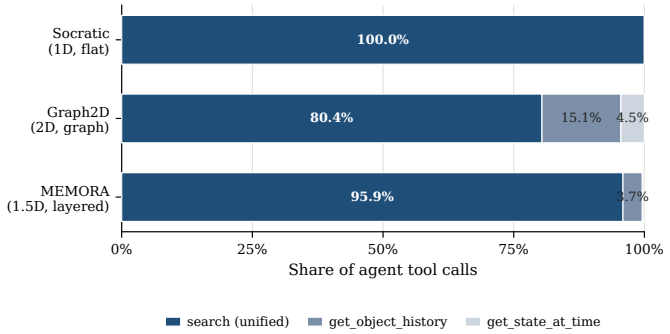

Fig. 17: **Tool-call distribution on EAM-QA across baselines and MEMORA.** Aggregated from 7,068 questions and 17,220 tool calls (4 backbones $\times$ four EAM-QA question types: ELOC, EORDER, SPREF, ERECALL). Socratic exposes only a single `search` (100%); Graph-2D adds time-localised helpers; MEMORA exposes a typed-dispatching `search` that routes to all four memory categories internally, plus two specialised tools (`get_object_history`, `get_state_at_time`). Unlike on planning (Figure 16), QA agents overwhelmingly use the unified dispatcher, reaching **Environment Memory** and the other typed indices through a single call.

c) 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.

d) Why no token-overlap metric cleanly favours 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 XVI: 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 favours 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	Replay	.912 / .883	$-.029$	.481 / .482	$+.001$	.477 / .461	$-.016$
(MoE, 4B active)	Generalize	.916 / .885	$-.031$	.663 / .610	$-.053$	.629 / .579	$-.050$
Gemma-4-31B-it	Replay	.904 / .901	$-.003$	.490 / .491	$+.001$	.490 / .491	$+.001$
(dense)	Generalize	.918 / .905	$-.013$	.679 / .645	$-.034$	.638 / .616	$-.022$
Qwen3.6-35B-A3B	Replay	.909 / .883	$-.026$	.491 / .483	$-.008$	.494 / .452	$-.042$
(MoE, reasoning)	Generalize	.925 / .892	$-.033$	.622 / .598	$-.024$	.608 / .566	$-.042$

This appendix documents the qualitative robot case study referenced in Section V-D and visualized in Figure 4. The case study checks whether the experience grounding measured by MEMORA-Planning remains 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, preference, temporal-order, and entity-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.

T. Setup and Protocol

a) 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.

b) Task design.: Each instruction is intentionally under-specified unless the system can use the participant's prior experience. **Task A: Prepare Drink** tests preference and entity grounding by checking whether the robot selects bottled Coca-Cola over a yellow Fanta pineapple can and the demonstrator's orange plastic cup among cups with other colors. **Task B: Breakfast** tests preference, grounding, and sequence by checking whether the robot selects the yellow bowl rather than a gray bowl, places Cinnamon Tea rather than Lipton Black Tea into the bowl, places the yellow spoon rather than a metal spoon into the bowl, and then takes the bread. The intended test is whether a memory system can transfer experience-derived context into a robot plan that resolves preferences, temporal order, entity identity, and attributes, rather than following a generic task script.

c) 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.

d) Execution interface.: MEMORA produces a language-level plan for each experience-grounded instruction. The generated plan is then converted into grasp and placement primitives by a fixed rule-based low-level controller. This design holds motor execution fixed, so the demonstration isolates memory-grounded language planning as the variable of interest.

U. Results and Traces

Table XVII gives the task-level plan and success criterion, and Figures 19 and 20 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 routines, preferences, and choices are transferred into robot behavior.

Task	Distractors / ambiguity	Generated MEMORA plan used for execution	Success criterion
PREPARE DRINK	Fanta can 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 selects the orange cup rather than other cups.
BREAKFAST	Multiple bowls, tea items, utensils, and side items are present, so both choices and order are experience-dependent.	(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 XVII: **Robot-transfer task specification and generated plans.** The plans are language-level MEMORA outputs used to drive a fixed rule-based controller.

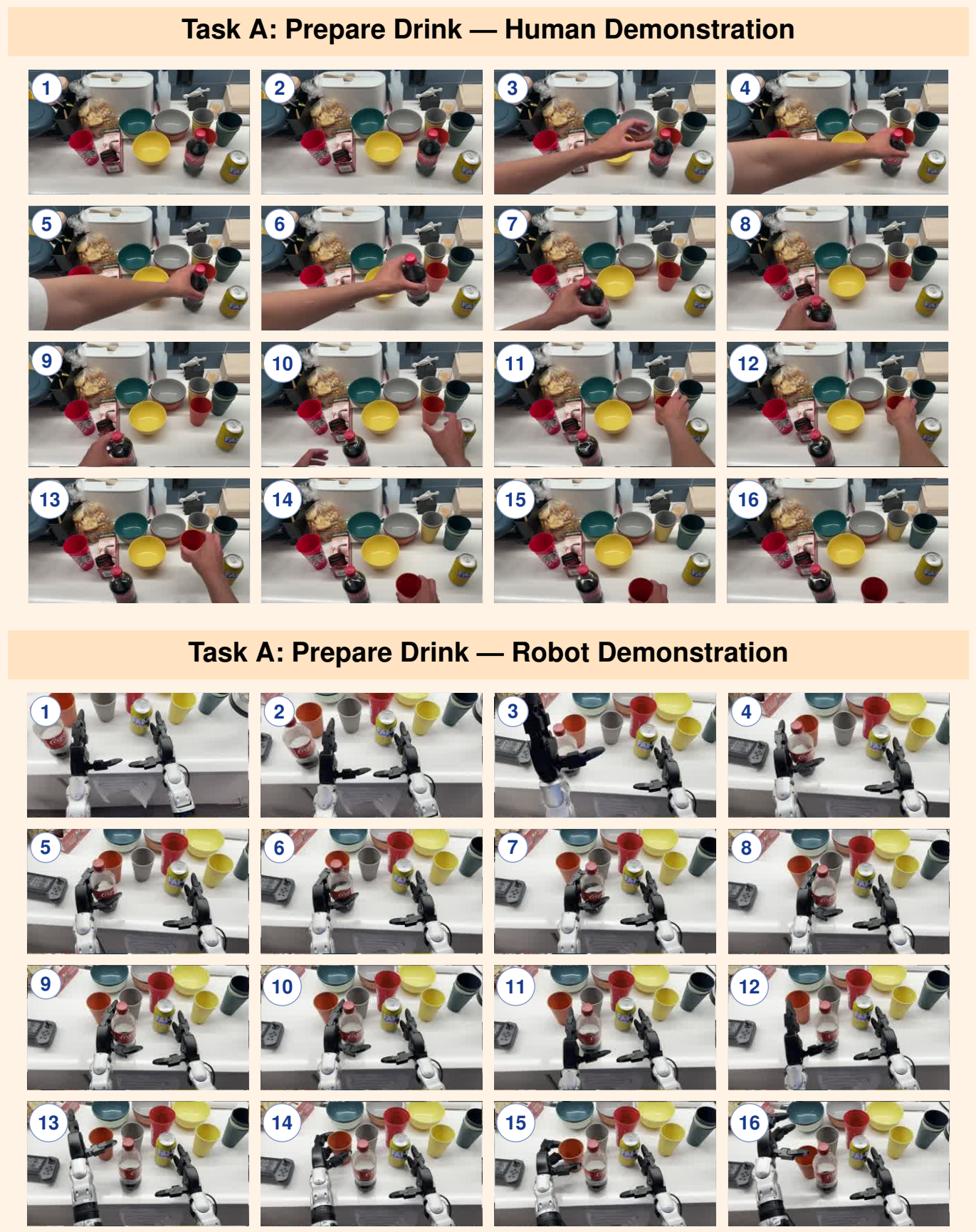

Fig. 19: **Task A: Prepare Drink (full trace).** The contact sheet shows the human egocentric demonstration and the Unitree G1 execution guided by MEMORA's memory-grounded language plan and a fixed rule-based controller. This task tests preference and entity grounding: success depends on selecting bottled Coca-Cola over a visually salient Fanta can and the demonstrator's orange plastic cup among multiple colored cups.

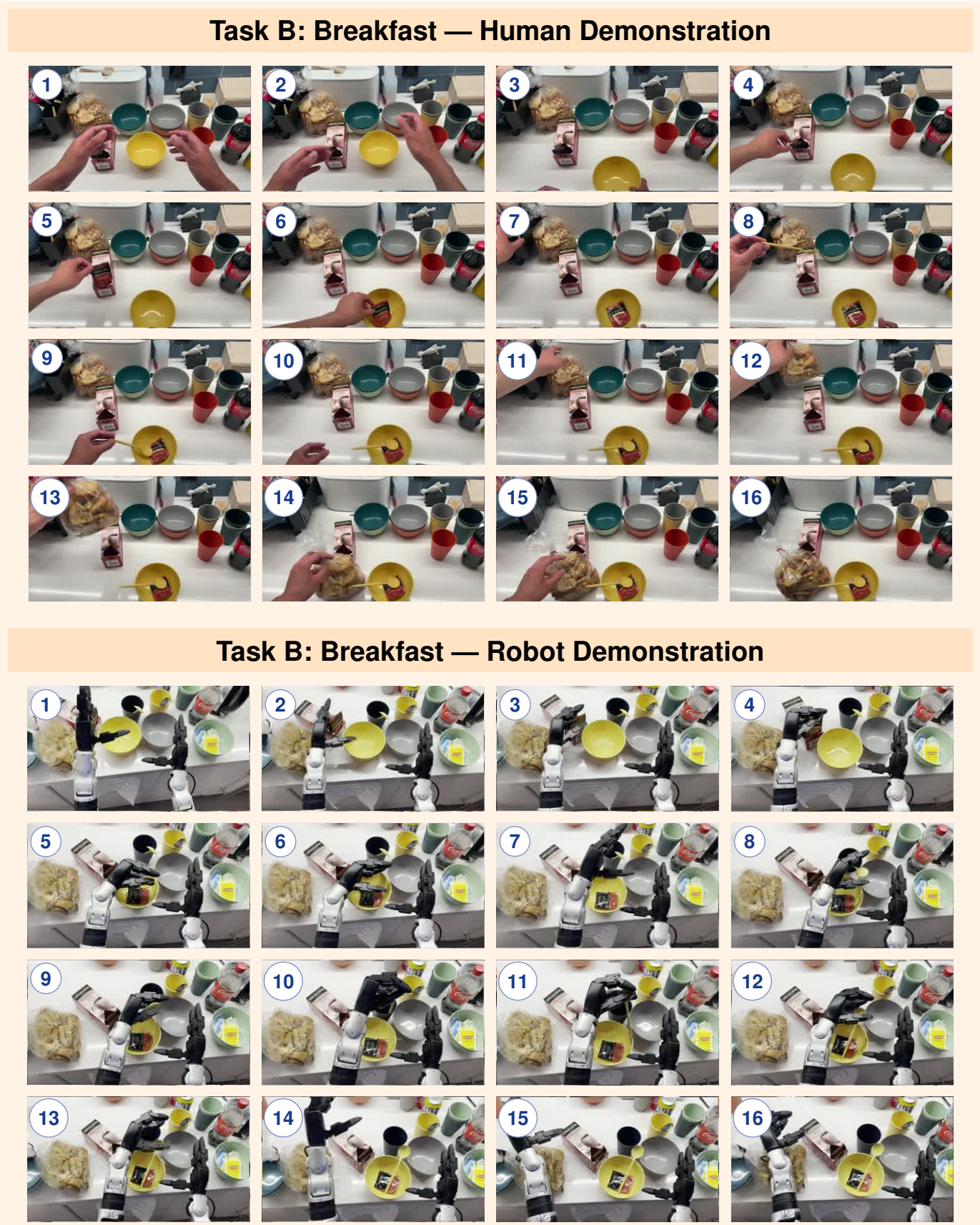

Fig. 20: **Task B: Breakfast (full trace).** The contact sheet shows the human egocentric demonstration and the Unitree G1 execution guided by MEMORA's memory-grounded language plan and a fixed rule-based controller. This task adds ordered procedural execution: success depends on selecting the yellow 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.

a) Limitations.: This demonstration is best 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 illustrating how memory formed from human egocentric observation can guide physical action selection, entity grounding, and sequencing in a robot execution setting.

Appendix U0a defines three orthogonal controlled ablations over the MEMORA memory-bank construction pipeline and its downstream answer agents. **CTRL** uses the main-paper processor settings in Appendix X0d (processor contracts: Table IV); 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, $\sim$80 EPIC-KITCHENS-100 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, $\sim$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 X0d and the **CTRL** row of Table XVIII. 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$\rightarrow$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.

V. Master Condition Matrix

a) Axis 1: Online Memory Processor.:

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

c) 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. CTRL in Table XVIII). 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 XX, to verify that the encoding-level conclusion is not specific to a particular answerer.

d) 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 XIX 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-rate surge in Table XX (Gemma block): temporal questions (SHABIT, SROUTINE) lose the most because they depend on cross-segment state-update density, not on the entity inventory.

e) Axis 2: Perception Processor and Audio.:

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

g) Design.: The Online Memory Processor is fixed at Qwen3-30B-A3B-Instruct-2507; we report results on two answer backbones in Table XX. **CTRL** 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 X0d: 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

TABLE XVIII: 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 **CTRL**; all other components match the **CTRL** settings in Appendix X0d. 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 CTRL therefore covers "audio off" as a side effect of the backbone change.

ID	Ax.	Changed component	Held fixed
CTRL	–	Main-paper CTRL: 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, Gemma-4-31B-it answer agents	Fixed CTRL consolidated **Inferred Knowledge** bank; reported in Table IX and Figure 15 (per-axis detail in Table XIV).

TABLE XIX: 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 CTRL (lower No-edit) while keeping a comparable entity inventory (similar MEMORA).

	CTRL (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.**

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. CTRL in Table XVIII).

h) Axis 3: Answer agent.:

i) 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?

j) 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 CTRL 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 (CTRL), Qwen3.6-27B, Gemma-4-26B-A4B-it, Gemma-4-31B} are scored on MEMORA-Bench EAM-QA in Table IX and on MEMORA-Planning in Figure 15 (eight cells over Replay ×

Generalize $\times$ four backbones; per-axis detail in Table XIV). 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 the project tool-call parsers without a standardized think mode.

k) 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.

W. 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 insufficient information) and per-type accuracy on SPREF/SROUTINE/SHABIT/ERECALL. Axis 3 is read off of the existing cross-backbone panels (Table IX, Figure 15) without re-running.

X. Results

a) Axes 1–2 results on two answer backbones (14 PIDs, 2,017 EAM-QA questions).: Table XX reports overall accuracy, E-rate (the rate at which the answer agent selects the explicit "E" *insufficient evidence* option), 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.

b) 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 is accompanied by a sharp rise in E-rate ($42.8\% \rightarrow 62.3\%$), indicating that the smaller model produces memory-bank records that less often support a contentful answer. Table XIX 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 answer agent selects E because the underlying state-change record is sparser, not because the bank is 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 essentially unchanged E-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).

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

d) Qualitative side-by-side comparison.: Figure 21 contrasts the three arms on two illustrative EAM-QA items, picked to expose the two ablation-specific failure modes implied by the

TABLE XX: 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 CTRL" is the overall-accuracy gap relative to **CTRL** (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 CTRL**	**SPref**	**SRoutine**	**SHabit**	**ERecall**
Answer backbone: Gemma-4-26B-A4B-it							
CTRL (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							
CTRL (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

aggregate accuracy and E-rate numbers in Table XX (Gemma block). On both items, CTRL (Omni + Qwen3-30B-A3B-Instruct-2507) recovers the gold letter; the memory-ablation arm A1-14B (Omni + Qwen3-14B) selects "E" – consistent with the $+19.5$ points E-rate inflation we report at the aggregate level; 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.

This section consolidates the model registry, inference stack, perception preprocessing, and retrieval/agent-loop settings that are shared across benchmark generation (Appendix 0a), Embodied Memory evaluation (Appendix G), Planning evaluation (Appendix M0b), and the controlled ablations (Appendix U0a). 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.

e) 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 MC5 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 V0k.

f) Licences, intended use, and release plan.: 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 [2] 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 V0k); 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 will be released

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
CTRL (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 memory state no longer supports a contentful answer.
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
CTRL (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 memory state again lacks enough retrievable support for a contentful answer.
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.

Fig. 21: **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 XX (Gemma block): *(i)* the memory-ablation arm (A1-14B) loses the contentful answer because the under-written memory provides insufficient retrievable support, raising E-rate; *(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.

under permissive research licences (planned: Apache-2.0 for code; CC BY-NC 4.0 for derived task data, inherited from the upstream dataset) following acceptance.

g) Model registry.: Table XXI 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 IV. 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 U0a), or NONE (MEMORA-Episodic ablation bank without **Inferred Knowledge** enrichment).

h) 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 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.

i) 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. 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 is kept on GPU during evaluation; the reported memory-tool latency therefore includes

TABLE XXI: 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

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` → `search_objects` → `search_activities` → `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 time spent in answer-agent generation rather than memory access (Table XXII). Since MEMORA forms memory offline before task execution and planning is issued before low-level control, this latency is compatible with the language-level planning interface studied here. At Embodied Memory evaluation time we additionally apply the time-restricted snapshot $\widetilde{\mathcal{M}}_{t_{\text{ask}}}$ (Section IV): entity states are rolled back to the `ask_turn_id` of the question, and offline **Inferred Knowledge** records that post-date t_{ask} are withheld so the agent must answer from observations strictly preceding the question.

TABLE XXII: 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