ECoMEM: Explicit Concept Memory for Memory-Dependent Robot Control

Anonymous authors
Paper under double-blind review

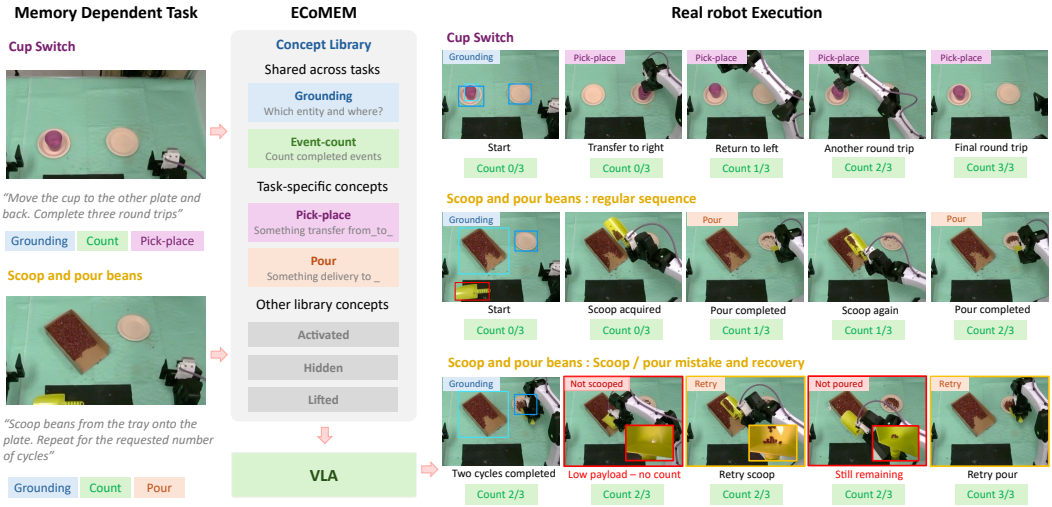


Figure 1: Explicit Concept Memory(ECoMEM) represents task-relevant history with compositional concept memory. **Left**: memory-dependent tasks require information that may no longer be available in the current observation. **Middle**: shared concepts for grounding and progress compose with task-specific event concepts. **Right**: the resulting memory conditions the VLA to track progress and recover from unsuccessful events on a real robot.

Abstract

A robot may lose sight of an object it must later retrieve, need to recall what a person demonstrated earlier, or track which steps of a task it has already completed. Current vision–language–action (VLA) policies often fail once the information needed for action disappears from the current observation, making memory critical for long-horizon robot behavior. Existing approaches typically provide longer histories or learn implicit memory from observation–action trajectories. But action supervision tells a policy *how to act*, not *what to remember*: it does not specify which past facts should persist or how they should change as new evidence arrives. We therefore separate *maintaining an evidence-grounded account of the past* from *learning how to act on it*. This insight motivates **Explicit Concept Memory (ECoMEM)**, which represents task-relevant history with a reusable library of grounded concepts. An evidence-based Writer selects and updates these records, while a learned Reader turns them into memory tokens that directly condition the VLA. Across 16 RoboMME tasks, ECoMEM leads the evaluated robot policies on 15 tasks. On two new real-robot tasks, the same memory library either transfers directly or requires only one new concept, achieving 86.1% success versus 8.6% for a no-memory VLA. These results show that explicit concepts provide a reusable and extensible memory interface for robot control.

1 Introduction

A robot that acts over extended periods cannot treat every observation as a fresh start. It may need to remember where an object was last seen after it becomes occluded, which object a person interacted with earlier, what parts of a task have already been completed, or the order in which previous events occurred. In each case, the information needed for the next action is no longer available in the current observation. Yet the relevant past is often much smaller than the full interaction history: what matters is a compact set of task-relevant facts about entities, states, events, and progress.

Vision–language–action (VLA) policies increasingly incorporate history through extended context, recurrent representations, learned memory banks, and test-time adaptation (Torne et al., 2025; Koo

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

et al., 2026; Shi et al., 2026; Jiang et al., 2026). These mechanisms can preserve rich perceptual history, but they leave an important question implicit: *what from that history should become memory?* Action supervision specifies what the robot should do, but not which past facts should be retained, when a belief should be revised, or when two observations correspond to the same event. When memory is learned only through the control objective, the policy may instead rely on shortcuts in current observations, previous actions, or trajectory-specific correlations (de Haan et al., 2019; Wen et al., 2020; Zhao et al., 2025). Successful action prediction therefore does not by itself establish that the task-relevant history has been represented reliably.

Making memory explicit helps, but does not remove the problem of memory construction. Keyframe memories preserve visual evidence (Yang et al., 2026; Fang et al., 2025), leaving the policy to recover the relevant facts from stored images. VLM-based approaches summarize histories (Sridhar et al., 2026; Anwar et al., 2025), but place entity tracking, state estimation, and temporal interpretation inside long-video reasoning, where spatial and temporal errors remain challenging (Qi et al., 2025; Seth et al., 2025). Textual world states (Yoneda et al., 2024; Liu et al., 2023) and scene graphs (Gu et al., 2024; Rana et al., 2023; Liu et al., 2025) provide more structured representations, but a robotic memory must additionally decide how observations become persistent facts: what should survive occlusion, what should be revised, what constitutes a completed event, and how repeated evidence should be reconciled.

Our key idea is to separate *maintaining an evidence-grounded account of the past* from *learning how to act on it*. Rather than asking the policy to infer memory implicitly from history, we represent task-relevant history as explicit facts about entities, states, completed events, progress, and temporal order. These facts are updated only when supported by visual and robot-state evidence, while the policy learns how they should influence action. This separation also keeps task requirements distinct from observations: the instruction specifies what should be achieved, while memory records what has actually happened.

We instantiate this principle in **Explicit Concept Memory (ECoMEM)**, an explicit memory channel for VLA policies (Figure 2). First, a shared concept library provides reusable memory primitives spanning entity and spatial grounding, state and relations, events and progress, and temporal and procedural structure. Second, an evidence-based *Writer* selects the concepts required by the instruction, grounds them to task entities, and maintains their records over time, while a learned *Reader* maps the resulting structured memory into latent tokens. Third, these memory tokens directly condition the VLA alongside visual and language inputs. Recognition and memory writing remain fixed during policy training, so evidence determines *what is remembered*, while action supervision learns *how that memory should be used*.

Across the 16 memory-dependent tasks of RoboMME (Dai et al., 2026), ECoMEM achieves the highest average success among the evaluated robot policies and leads on 15 of 16 tasks. Controlled experiments show that spatial grounding, event and progress information, temporal structure, task-conditioned selection, and language grounding each contribute to effective memory construction. More importantly, the same abstraction transfers to new physical-robot tasks: CUPSWITCH reuses the existing library without adding a new recognizer, while SCOOPPOUR requires only one new event concept composed with the existing memory machinery. Across shared real-robot conditions, ECoMEM succeeds in $68/79$ trials (86.1%), compared with $5/58$ (8.6%) for the no-memory $\pi_{0.5}$ baseline. These results suggest that explicit, compositional concepts can serve not merely as task-specific memory annotations, but as a reusable and extensible interface between a robot's accumulated experience and learned control.

2 RELATED WORK

2.1 IMPLICIT MEMORY IN ROBOT POLICIES

History-aware robot policies typically learn a latent representation of past observations and actions. Long-context policies directly attend to extended trajectories (Torne et al., 2025; Koo et al., 2026); recurrent architectures propagate learned state across time (Li et al., 2026); memory-bank methods compress and retrieve past perceptual features (Shi et al., 2026); and test-time adaptation stores history in model weights (Jiang et al., 2026). Predictive world models similarly maintain latent state for control (Wu et al., 2023; Samsami et al., 2024). These approaches can capture rich temporal dependencies without prescribing what the representation must contain. However, what is retained, forgotten, or updated is learned largely through downstream objectives and remains difficult to inspect independently of the policy. ECoMEM instead externalizes a sparse set of task-relevant facts

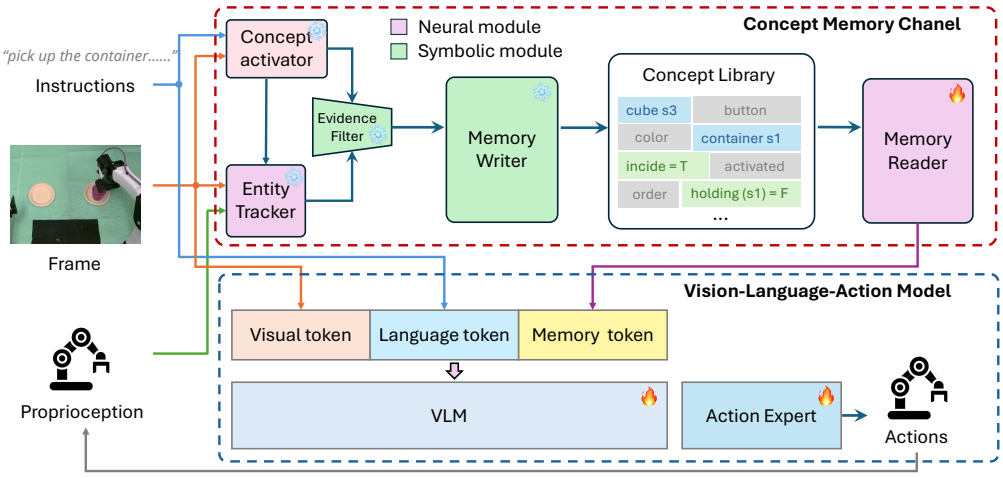


Figure 2: ECoMEM separates memory construction from memory use. The Writer selects task-relevant concepts, recognizes them from visual and robot-state evidence, and maintains structured records. The learned Reader encodes these records as memory tokens that condition the VLA alongside image and language tokens.

and fixes their evidence-based updates, while learning only how the policy should use them. Explicit and latent memory are therefore complementary rather than mutually exclusive.

2.2 Explicit Memory in Robot Policies

A second line of work stores selected parts of experience explicitly. Keyframe and retrieval methods retain event-triggered images or trajectory segments (Yang et al., 2026; Fang et al., 2025; Memmel et al., 2025; Jain et al., 2024); MemER retrieves task-relevant visual experiences and uses a high-level VLM to condition a low-level policy (Sridhar et al., 2026). Other approaches summarize experience into language (Anwar et al., 2025; Liu et al., 2023), maintain textual world states (Yoneda et al., 2024), or organize persistent information in scene graphs (Gu et al., 2024; Rana et al., 2023; Liu et al., 2025). These representations improve access to past information, but visual memories still require the controller to recover facts from stored observations, while language and graph memories are commonly consumed by high-level reasoning or planning modules. ECoMEM instead maintains typed, evidence-linked records of states, events, progress, and temporal structure, and feeds them directly to a low-level VLA through a learned Reader.

2.3 Neuro-Symbolic Methods in Robotics

Neuro-symbolic robotics combines learned perception and control with explicit, compositional abstractions. Knowledge representations such as KnowRob organize persistent robot knowledge (Tenorth & Beetz, 2009); neuro-symbolic concept learning and structured world models ground symbolic concepts in perception and compose them for reasoning (Mao et al., 2019; 2022); and learned or invented predicates provide compact abstractions for task planning (Liang et al., 2025; Athalye et al., 2026; Silver et al., 2023). Related work also uses symbolic concepts to structure learned generative models for manipulation and geometric reasoning (Xu et al., 2024; 2025). Recent systems further use vision–language models to construct symbolic planning abstractions from demonstrations (Lorang et al., 2026), while knowledge graphs provide structured memory for language-model agents (Anokhin et al., 2025). ECoMEM follows the same principle of grounding reusable symbolic concepts in learned representations, but uses them for a different role: as the interface between perceptual history and a learned control policy.

3 Preliminaries

Memory-dependent tasks. Given an instruction ℓ, a VLA policy observes o_t (camera images and robot state) at decision t and predicts an action chunk a_t. The history h_t contains all observations and executed actions up to t, plus any demonstration given with the task. A task is *memory-dependent* if two histories with the same instruction and current observation require different expert actions:

$$\pi^\star(\cdot \mid \ell, h_t) \neq \pi^\star(\cdot \mid \ell, h'_t) \quad \text{with } o_t = o'_t. \tag{1}$$

Pressing a button three times illustrates this: the scene may look identical after the first and third press, yet the correct action differs. A policy conditioned only on (ℓ, o_t) cannot distinguish them.

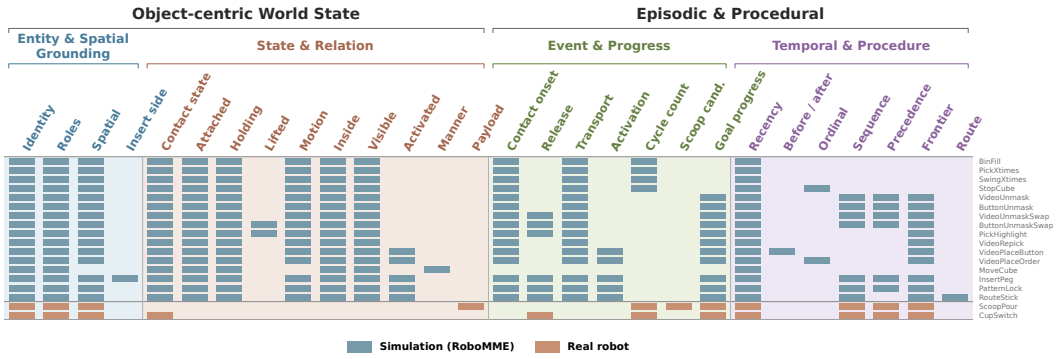


Figure 3: A shared concept library supports diverse memory tasks. Columns show concepts grouped into four semantic families, and rows show the 16 RoboMME tasks and two real-robot tasks. Filled cells indicate the concepts used by each task, revealing extensive reuse of shared concepts across otherwise different memory demands.

Memory-conditioned policy. The history is long, but what separates h_t from h'_t is usually a few facts. We therefore compress the history into a memory m_t: a set of concept records, each stating one fact about the task entities (Figure 3). We separate what to remember from how to use it:

$$m_t = W(\ell, h_t), \qquad u_t = R_\phi(m_t), \qquad a_t \sim \pi_\theta(\cdot \mid \ell, o_t, u_t). \tag{2}$$

We construct a Writer W that selects concepts, harnesses pretrained perception models to recognize them, and updates records using only the history available at t (Section 4.2). We keep W fixed, because action supervision alone does not specify which facts to keep; a learned memory can instead exploit shortcuts (de Haan et al., 2019; Wen et al., 2020). We then train a Reader R_ϕ jointly with the VLA π_θ on expert action chunks $a_t^\star$. The Reader encodes the records into memory tokens u_t that condition the policy (Section 4.3, Appendix A).

4 ECoMEM

ECoMEM adds an explicit memory channel to a VLA (Figure 2), following the factorization in Equation 2. The *Writer* W determines what to remember: it draws from a shared concept library (Section 4.1), selects the concepts required by the task, recognizes them from visual and robot-state evidence, and updates the memory records (Section 4.2). The *Reader* R_ϕ learns how to use this memory by encoding the records into tokens that condition the VLA (Section 4.3).

4.1 Concept Library

A shared concept library. Memory-dependent tasks differ, but the facts they need recur. Counting button presses and counting placements both need the number of completed events. Finding a hidden cube and re-picking a demonstrated object both need a remembered location. We therefore define each type of fact once, as a concept in a shared library. A new task reuses these concepts instead of building from scratch, and a single Reader serves all tasks. Robot knowledge bases and neuro-symbolic concept learners follow the same principle (Tenorth & Beetz, 2009; Mao et al., 2019).

Four concept families. We keep a concept only if it can change the correct action when the instruction and current observation are the same (Equation 1). The concepts that meet this criterion fall into four families (Figure 3). Each family resolves an ambiguity that the previous ones cannot. **Entity & Spatial Grounding** answers *what and where*. A remembered fact is useful only if it stays attached to the right object, so this family binds facts to persistent entities and their locations (Gu et al., 2024). **State & Relation** answers *in what state*. Knowing the target does not tell the robot what to do with it, so this family records states and relations such as `holding` and `inside` (Silver et al., 2023). **Event & Progress** answers *what has happened*. The same scene can call for continuing or stopping, so this family records completed events and progress toward the goal (Huang et al., 2023). **Temporal & Procedure** answers *in what order*. Counts ignore order, so this family records the order of events and the required order of steps. The first two families describe the current world; the last two describe the past.

Concepts and records. We construct each concept with a coding agent. The library is built once and then reused in most of the other tasks. It specifies its arguments, a recognizer that harnesses pretrained perception models and robot state, and an update rule written as explicit logic. The library

stores concepts; memory stores their instances as records. For example, the record `INSIDE(s2, s1)` states that entity `s2` is inside container `s1`, with a truth value, a status, a time, and its supporting evidence. Concepts also compose. The pick–place concept builds on the transport concept, which in turn builds on contact and region visits.

Extending the library. We build the library to grow with new tasks. Because concepts compose, a new concept reuses the existing perception models, lower-level concepts, and record format. An agent constructs the new concept on top of the library, and we verify it on a few demonstrations (Appendix I.2). For example, scooping beans into a bowl twice needs a scooping concept, which the library lacks. We could add `SCOOP` to the library: it reuses entity grounding, containment, and event counting, and adds only a check that the scooper carries beans (Section 5.5).

4.2 Constructing Concept Memory

The Writer W builds the memory m_t in three steps: it selects the concepts a task needs, recognizes them from the history h_t, and updates the records.

Task-conditioned selection and grounding. Running every concept is costly, and it adds records the task does not need, which distract the policy (Section 5.4). The Writer therefore activates only the concepts the instruction requires. A fixed instruction parser returns the required concepts $S(\ell, \mathcal{L})$, and $\mathrm{DepClosure}$ adds the concepts they depend on. For each active concept, Inst extracts its parameters η_c from the instruction:

$$\mathcal{C}_\ell = \mathrm{DepClosure}\big(S(\ell, \mathcal{L})\big), \qquad \eta_c = \mathrm{Inst}(\ell, c), \quad c \in \mathcal{C}_\ell. \tag{3}$$

The parameters η_c include entity phrases for *objects*, *destinations*, *controls*, and also the required properties, such as *counts*, *ordinals*, and *order constraints* (Appendix C). These two operations output answers to different questions: $\mathcal{C}_\ell$ decides which concepts to recognize, and η_c decides which entities each concept grounds, so each record binds to the object the task refers to rather than to any object in view. Task conditioning only decides what to observe but does not establish facts: *repeat three times* sets a desired count, and only observed completion events increase the completed count.

Multimodal concept recognition. No single modality recognizes a concept reliably. Vision finds objects but cannot tell whether the gripper holds one; robot state tells when the gripper closes but not on what. Each recognizer therefore composes evidence across modalities. We harness OWLv2 (Minderer et al., 2023) to ground the entity phrases as bounding boxes, and SAM2 (Ravi et al., 2025) to track them across frames where needed. End-effector positions and gripper states show when and where the robot acts, and camera calibration projects the gripper into the image, so visual and robot-state evidence can be compared directly. Each active concept c runs its recognizer $\mathrm{Rec}_c(h_t; \eta_c)$, which composes these signals at three levels. Spatial and relational checks describe how entities are arranged at one time. Event checks test changes over an interaction interval. Temporal operators assemble confirmed events into counts or ordered sequences. Each result keeps its entity bindings, evidence, and time.

Evidence merging and memory update. Recognition outputs are noisy and repetitive. The same event appears in many frames, and objects disappear behind occluders. The Writer merges the outputs of all active concepts and updates the previous memory:

$$m_t = \mathrm{Update}\Big(m_{t-1},\ \big\{\mathrm{Rec}_c(h_t; \eta_c)\big\}_{c \in \mathcal{C}_\ell}\Big). \tag{4}$$

Update turns noisy outputs into stable memory with four rules. First, a state needs accumulated evidence before it is confirmed, and a missed detection is not evidence against it; an occluded object therefore keeps its last location. Second, an event is recorded once, when it is first confirmed; seeing it again adds nothing. Third, a count is the number of distinct recorded events. Fourth, progress compares counts with the goal, and an ordered step counts only after its prerequisites are complete. All updates use only observations up to the current decision. A demonstration given with the task can set entity bindings and the required order, but its events never count as the robot's own progress.

4.3 Integrating Concept Memory with VLA

The Reader R_ϕ turns the memory m_t into tokens u_t that the VLA π_θ can attend to.

Reading memory records as tokens. Each record $r \in m_t$ is a tuple of K discrete fields: kind, predicate, subject and object entities, truth value, status, confidence, age, progress flags, and grid location (Appendix F). Writing records as text would flatten this structure and lengthen the prompt, so the Reader embeds the fields directly. Each field k has its own embedding table E_k, because the same value means different things in different fields. Summing the field embeddings gives one vector $e(r)$ per record. A small Transformer T_ϕ then relates the records and maps each one to the VLA's token width:

$$e(r) = \sum_{k=1}^{K} E_k(r_k), \qquad u_t = R_\phi(m_t) = T_\phi\big(\{e(r)\}_{r \in m_t}\big). \tag{5}$$

The Reader's parameters ϕ are the tables E_k and the Transformer T_ϕ. The Reader outputs one token per record, without pooling, so no fact is averaged away. Attention across records lets it relate facts, such as a count and its goal. Records form a set, so the Reader uses no position embeddings; temporal information enters through the age field and goal dependencies.

Policy integration and training. The tokens u_t join the image and language tokens in the prefix of π_θ, which we initialize from pretrained $\pi_{0.5}$ (Physical Intelligence et al., 2025). The pretrained policy has never seen memory tokens, so it must learn how they relate to actions. We therefore train ϕ jointly with the language backbone and action expert of π_θ, using the base model's flow-matching objective. The vision encoder and the Writer W stay frozen. We compute training memory as at execution: m_t uses only observations up to decision t. Entity identifiers are arbitrary labels, so we randomly permute them in each training sample; the policy cannot rely on specific numbers. At execution, W updates m_t as observations arrive, and R_ϕ reads it before each action chunk.

5 Experiments

Our experiments address four questions. **(1)** How does explicit concept memory compare with no-memory, history-based, and visual-memory alternatives (Section 5.2)? **(2)** Do different concept families provide distinct information needed for memory-dependent decisions (Section 5.3)? **(3)** What is required to construct useful concept memory, and how important are structured recognition, task-conditioned selection, and language grounding (Section 5.4)? **(4)** Can the concept library transfer to new real-robot tasks through reuse and extension (Section 5.5)?

5.1 Experiment Setup

Tasks. We evaluate on the 16 memory-dependent tasks of RoboMME (Dai et al., 2026) and two new real-robot tasks. RoboMME covers four memory demands: counting repeated events, maintaining objects through occlusion, referring to past interactions, and reproducing demonstrated behavior. On the real robot, **CupSwitch** requires completing a requested number of cup round trips between two plates, while **ScoopPour** requires a requested number of successful scoop-and-pour cycles; empty scoops do not count. These tasks introduce new objects and success conditions, and ScoopPour additionally requires a concept absent from the base library. We use a 7-DoF YAM arm and collect demonstrations with GELLO (Wu et al., 2024); further platform details are in Appendix I.

Baselines. On RoboMME, we compare against the baselines reported by RoboMME: $\pi_{0.5}$ without memory, $\pi_{0.5}$ with past actions, FrameSamp+Modul, SAM2Act+, and MemER. Together they cover no memory, history conditioning, and visual-memory approaches. We additionally compare the Writer with a fine-tuned VLM that extracts the same concepts directly from video (Section 5.4). On the real robot, we compare against $\pi_{0.5}$ fine-tuned on the same demonstrations without memory.

Training and evaluation. We initialize ECoMEM from pretrained $\pi_{0.5}$ and jointly train on all 16 RoboMME tasks for 200,000 updates. Each task is evaluated on 50 validation episodes with three policy sampling seeds, and we report per-task success and the task-equal mean. Controlled ablations use matched 50-episode evaluations and their own full-memory controls; their absolute scores are therefore not directly comparable with Table 1 (Appendices A and B). For the real robot, both methods are fine-tuned from pretrained $\pi_{0.5}$ on the same teleoperated demonstrations, and success requires completing exactly the requested number of cycles (Appendix I).

5.2 Concept Memory Outperforms Implicit and Visual Memory

Explicit concept memory substantially improves memory-dependent control. ECoMEM achieves 82.42% average success, 37.91 points above the strongest baseline, FrameSamp+Modul,

Table 1: Success rates (%) on the 16 RoboMME memory-dependent tasks. AVG is the task-equal mean across all tasks. Baseline and human results are reported by RoboMME (Dai et al., 2026).

Method	Counting				Permanence				Reference				Imitation				AVG
	Bin Fill	Pick Xtimes	Swing Xtimes	Stop Cube	Video Unmsk	Button Unmsk	Video UnmskS	Button UnmskS	Pick HighL	Video Repick	Video PicBtn	Video PicOrd	Move Cube	Insert Peg	Pattern Lock	Route Stick	
Human Performance	**96.00**	**100.00**	**80.00**	**78.00**	**90.00**	**92.00**	**92.00**	**90.00**	**92.00**	**92.00**	**98.00**	**90.00**	**90.00**	**98.00**	**84.00**	**86.00**	**90.50**
FrameSamp+Modul	39.56	87.33	92.00	42.00	32.67	25.11	24.44	18.22	22.89	30.44	60.00	32.00	77.78	7.56	53.56	66.67	44.51
$\pi_{0.5}$	30.00	42.89	35.56	6.67	20.44	22.22	18.67	6.67	11.33	0.44	31.11	25.78	26.00	1.56	2.89	4.67	17.93
$\pi_{0.5}$ w/ past actions	26.67	58.33	26.67	4.67	30.67	23.67	20.67	16.00	12.33	8.67	24.00	18.67	34.00	1.00	4.00	5.67	19.73
SAM2Act+	40.00	76.00	25.33	0.00	27.33	32.00	18.00	26.67	17.33	5.33	24.67	20.00	29.33	0.00	0.00	0.00	21.37
MemER	56.67	79.33	59.33	0.00	81.33	72.00	38.00	21.33	**70.67**	25.33	30.00	26.00	82.67	6.67	16.67	12.00	42.38
ECoMEM	**71.33**	**100.00**	**99.33**	**66.00**	**100.00**	**96.67**	**78.00**	**84.00**	**62.67**	**77.33**	**96.00**	**86.00**	**96.00**	**40.67**	**79.33**	**85.33**	**82.42**

and leads the evaluated policies on 15 of 16 tasks (Table 1). Its average is within 8.08 points of human performance.

The advantage is largest when the required information must persist beyond the visible frame. The largest margins occur on ButtonUnmaskSwap, VideoPlaceOrder, and VideoRepick, where the robot must act on a target, location, or ordering established earlier but no longer directly observable. Here, ECoMEM preserves entity bindings, locations, and event order as persistent records rather than requiring the policy to recover them from visual history. Counting tasks similarly benefit from explicit completion events and progress. By contrast, gains are smaller on SwingXtimes and MoveCube, where sampled or retrieved frames already expose much of the relevant history. PickHighlight is the only task where ECoMEM does not lead: MemER is 8 points higher, consistent with a transient visual cue being directly preserved in a retrieved keyframe.

Remaining errors expose control limits beyond memory. On InsertPeg, ECoMEM reaches 40.67%, compared with 7.56% for the strongest baseline, but remains well below human performance. Memory provides the insertion side and endpoint; precise execution must still be learned by the policy. Section 5.3 next isolates which information in concept memory drives these gains.

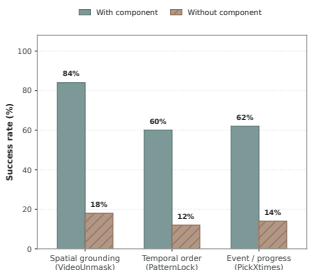


Figure 4: Different memory decisions depend on complementary concept families. Removing spatial grounding, temporal order, or event and progress information sharply reduces success on the corresponding memory-dependent task. Each pair is trained and evaluated with the same intervention on 50 matched episodes.

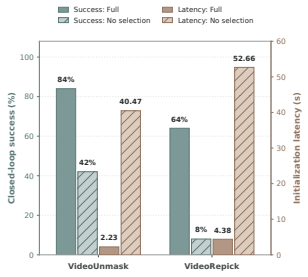


Figure 5: Task-conditioned concept selection improves both control and efficiency. With the policy fixed, full-library recognition adds task-unselected records, reduces success on VideoUnmask and VideoRepick, and increases memory initialization latency by 12.0× to 18.1×. Latency is measured on matched episodes using the same GPUs.

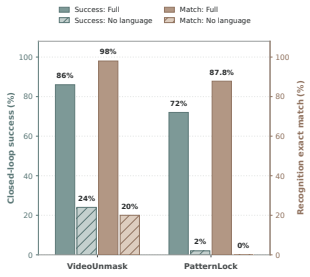


Figure 6: Language-conditioned grounding binds memory to the correct entities. Replacing instruction-specific entity phrases with generic role-level queries sharply reduces both reference-record agreement and closed-loop success on VideoUnmask and PatternLock. Concept selection and the policy are fixed.

5.3 Concept Families Provide Complementary Memory

A compact library is shared across diverse memory tasks. As shown in Figure 3, the 16 RoboMME tasks reuse concepts from the same four families rather than requiring task-specific memory representations. Tasks may differ in which concepts they activate and how they compose them.

Different memory decisions depend on different concept families. We remove spatial information on VideoUnmask, event and progress information on PickXtimes, and temporal order on PatternLock, while preserving the remaining memory and removing indirect cues to the ablated information (Appendix G.1). Each intervention is applied during both training and evaluation on matched data, allowing the policy to exploit any information that remains. Success drops from 84% to 18%, 62% to 14%, and 60% to 12%, respectively (Figure 4; exact McNemar $p < 0.002$ for all three). The failures align with the role of each family: *where* to act after occlusion, *how many* events have occurred, and *in what order* to act. Together, the coverage in Figure 3 and these targeted ablations show that a small shared vocabulary can support diverse memory demands while preserving the distinctions needed for control.

Table 2: Concept extraction across all four memory families. We evaluate eight whole-video concepts on 95 RoboMME test videos. Structured and QA are separately fine-tuned Qwen3-VL-8B-Instruct models; Concept Memory uses our evidence-based Writer. Cell shading encodes extraction accuracy (darker is higher); **bold** denotes the best result in each row. See Appendix H for protocols and limitations.

Family	Concept extracted	Structured	QA	**Concept Memory**
Entity & Spatial	Picked-object identity	71.4%	71.4%	**100.0%**
	Highlighted-object set	63.2%	63.2%	**97.4%**
State & Relation	Currently highlighted	**100.0%**	97.4%	97.4%
	Currently holding queried object	**100.0%**	96.7%	**100.0%**
Event & Progress	Pick–place cycle count	86.7%	93.3%	**100.0%**
	Region-entry count	**95.0%**	90.0%	**95.0%**
Temporal & Procedure	Ordered object–destination sequence	11.8%	5.9%	**100.0%**
	Placements before/after press	23.5%	23.5%	**88.2%**

5.4 Reliable Memory Requires Structured Construction

The Writer must do more than preserve history: it must extract the right structure, select what is relevant, and bind records to the right entities. We test each requirement against a simpler alternative. For structured extraction, we compare Concept Memory with fine-tuned VLMs on concepts spanning all four memory families. For selection and grounding, we intervene on the Writer at evaluation while keeping the trained policy fixed and using matched full-memory controls.

Direct VLM extraction is strong on local state, but weak on structured history. We fine-tune Qwen3-VL-8B-Instruct to extract each concept directly from video, either as a structured record or as an answer to a question, and evaluate eight concepts on 95 held-out RoboMME videos (Table 2). The VLMs recover current state and simple event counts reliably: state-and-relation concepts reach 96.7–100%, and event counts reach 86.7–95.0%. Performance drops for episodic entity information (63.2–71.4%) and collapses on temporal composition, reaching only 5.9–11.8% for ordered object–destination sequences and 23.5% for placements around an event. The similar Structured and QA results indicate that output format is not the main bottleneck. In contrast, Concept Memory achieves 95–100% on the first three families and 88.2–100% on temporal concepts. Explicitly composing grounded evidence into events and temporal records therefore provides a substantially more reliable memory construction mechanism than the tested direct video extraction.

Task-conditioned selection improves both accuracy and efficiency. *No Selection* runs every recognizer and adds all resulting records while retaining the task-relevant ones. Success drops from 84% to 42% on VideoUnmask and from 64% to 8% on VideoRepick, while median memory initialization becomes $18.1\times$ and $12.0\times$ slower (Figure 5). Thus, more memory is not necessarily better: task-conditioned selection reduces recognition cost and prevents irrelevant records from disrupting the policy.

Without language grounding, records bind to the wrong objects. *Generic queries* replace task-specific entity phrases with role-level ones such as *an object*, while keeping the selected concepts, detection budget, and policy instruction fixed. The Writer runs the right concepts but no longer knows which entities they refer to. Agreement with reference records falls from 98% to 20% for target locations in VideoUnmask and from 87.8% to 0% for ordered location sequences in PatternLock; closed-loop success falls from 86% to 24% and from 72% to 2% (Figure 6). The policy still receives the full instruction, yet it cannot repair memory records grounded to the wrong objects.

Together, these results show that useful memory depends on how it is constructed. Reliable memory requires structured evidence composition, task-conditioned selection, and language-grounded entity binding; the policy cannot reliably recover information that the Writer constructs incorrectly.

5.5 The Library Transfers to New Tasks on a Real Robot

The real robot tests whether the library transfers beyond RoboMME to new cameras, objects, and tasks. We consider two forms of transfer: **CupSwitch** tests whether a new task can reuse the existing library, while **ScoopPour** tests whether a missing capability can be added compositionally. Both policies initialize from pretrained $\pi_{0.5}$ and use the same real-robot demonstrations; ECoMEM additionally receives concept memory.

Table 3: Real-robot success (successes/trials) by requested number of cycles; success requires exactly that count. Pooled uses the targets both policies were run on.

	ScoopPour			CupSwitch		Pooled	
Method	$1\times$	$2\times$	$3\times$	$2\times$	$3\times$	succ./trials	rate
ECoMEM	16/20	19/23	11/12	11/12	11/12	**68/79**	**86.1%**
$\pi_{0.5}$ (no memory)	4/12	0/12	0/11	0/12	1/11	5/58	8.6%

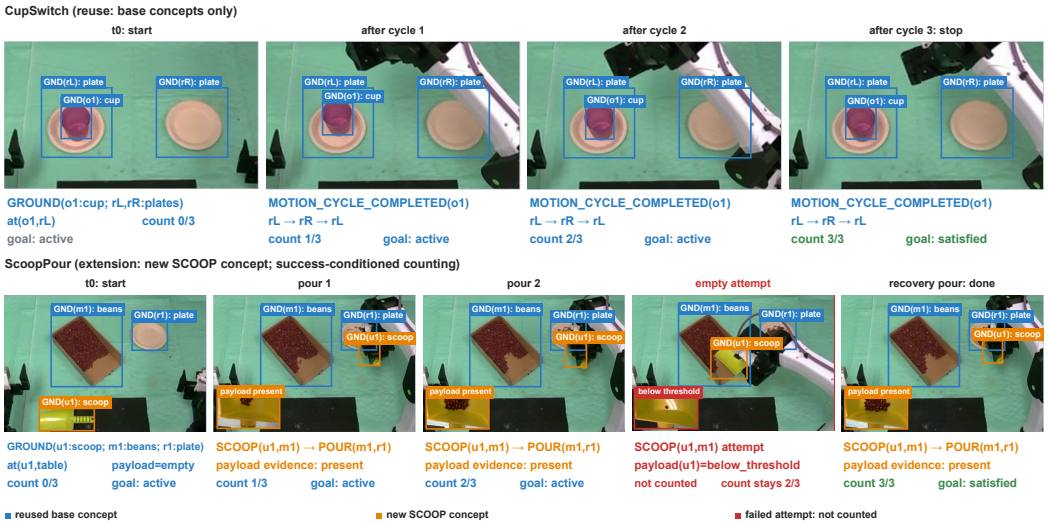


Figure 7: Concept memory transfers through reuse and compositional extension on a real robot. Blue: reused base concepts; orange: the new SCOOP concept. Both tasks require counting, but ECoMEM represents progress at the task-relevant granularity: CupSwitch advances after each completed round trip, while ScoopPour composes SCOOP with the same counter so that only successful scoops carrying beans advance progress. An empty scoop therefore leaves the count unchanged and automatically triggers a retry. Records are produced online by the Writer.

Transfer occurs through reuse and local extension. CupSwitch uses only existing concepts for grounding, persistence, pick-place cycles, event counting, and count goals, with no new recognizer or update rule (Figure 3). ScoopPour adds a single concept, SCOOP(tool, material), which confirms a scoop only when the lifted scooper carries beans. Grounding, persistence, counting, and goal tracking are reused unchanged. Thus, one new task is covered directly, while the other extends the library with one concept that composes with the existing machinery.

Conditioning on explicit memory is critical for repeated tasks. ECoMEM succeeds in 80% to 92% of ScoopPour trials and 92% of CupSwitch trials, remaining high across requested counts (Table 3). Without memory, $\pi_{0.5}$ reaches 33% on one ScoopPour cycle but 0% on two and three, and reaches 0% and 9% on two and three CupSwitch round trips. Across shared evaluation conditions, ECoMEM succeeds in 68/79 trials (86.1%), compared with 5/58 (8.6%) for the no-memory policy (Fisher exact $p < 10^{-20}$; trials are unpaired and unequal in number). The two systems share the pretrained backbone and demonstrations, while only ECoMEM conditions the policy on explicit memory tokens. The collapse of the no-memory policy as counts increase shows that current observations alone are insufficient once task progress is no longer visible.

Neural-symbolic concepts let memory adapt its granularity to the task. Because concepts compose learned perceptual recognition with explicit event and progress structure, the same memory machinery can track different notions of task progress. Figure 7 shows this online. In CupSwitch, the counter advances after each completed round trip, distinguishing visually similar scenes by how many cycles have occurred. In ScoopPour, SCOOP adds a success condition, so only scoops that actually carry beans advance the same counter; an empty scoop leaves progress unchanged and the policy retries. Thus, the library can represent progress at the semantic granularity each task requires by composing shared counting machinery with task-relevant concepts.

6 Conclusion

We introduced ECoMEM, an explicit concept memory interface that separates evidence-based memory construction from learned action generation. A shared, compositional concept library represents task-relevant world state, events, progress, and temporal structure, while the Writer maintains these records and the Reader learns how to use them for control. Across 16 RoboMME tasks, ECoMEM substantially outperforms no-memory, history-based, and visual-memory alternatives. Controlled ablations show that distinct concept families provide complementary information, and that structured recognition, task-conditioned selection, and language grounding are critical to constructing effective memory. On new real-robot tasks, the same library transfers through direct reuse or the addition of a single task-specific concept, showing that explicit concepts can provide a reusable and extensible memory interface for VLA control.

AI USE STATEMENT

The core idea of ECoMEM was proposed by the authors, who established the research objectives and led the methodological development, experimental investigation, and interpretation of results. Under author direction, generative AI tools assisted with refining technical details and experimental protocols, implementing and debugging code, preparing and analyzing data, searching the literature, and preparing figures. AI agents also assisted with concept-recognizer construction and development-video reference annotation, as described in Appendix I.2. For manuscript preparation, AI assistance was used primarily to improve language, clarity, and organization. The authors directed the revision process and determined the final content, claims, and wording. AI-assisted implementations and analyses were checked through code inspection, tests, and experimental records; selected development annotations received manual review. The authors retained final decision-making authority throughout the research and take full responsibility for the accuracy, originality, and integrity of the submitted work, including all AI-assisted materials.

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. AriGraph: Learning knowledge graph world models with episodic memory for LLM agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2025. doi: 10.24963/ijcai.2025/2.

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. ReMEmbR: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2409.13682.

Ashay Athalye, Nishanth Kumar, Tom Silver, Yichao Liang, Jiuguang Wang, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. From pixels to predicates: Learning symbolic world models via pretrained VLMs. *IEEE Robotics and Automation Letters*, 11(4):4002–4009, 2026. URL https://ieeexplore.ieee.org/document/11373861.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. RoboMME: Benchmarking and understanding memory for robotic generalist policies. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.04639.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. SAM2Act: Integrating visual foundation model with a memory architecture for robotic manipulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15925–15942, 2025. URL https://proceedings.mlr.press/v267/fang25c.html.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028, 2024. URL https://concept-graphs.github.io/.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782, 2023. URL https://proceedings.mlr.press/v205/huang23c.html.

Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2Robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.12943.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, and Linxi Fan. RoboTTT: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026.

Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. HAMLET: Switch your vision-language-action model into a history-aware policy. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.00695.

Hang Li, Fengyi Shen, Dong Chen, Liudi Yang, Xudong Wang, Jinkui Shi, Zhenshan Bing, Ziyuan Liu, and Alois Knoll. ReMem-VLA: Empowering vision-language-action model with memory via dual-level recurrent queries. *arXiv preprint arXiv:2603.12942*, 2026.

Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B. Tenenbaum, Tom Silver, João F. Henriques, and Kevin Ellis. VisualPredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.23156.

Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. DynaMem: Online dynamic spatio-semantic memory for open world mobile manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2411.04999.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484, 2023. URL https://proceedings.mlr.press/v229/liu23g.html.

Pierrick Lorang, Johannes Huemer, Timothy Duggan, Kai Goebel, Patrik Zips, and Matthias Scheutz. Build on priors: Vision–language–guided neuro-symbolic imitation learning for data-efficient real-world robot manipulation. *arXiv preprint arXiv:2604.03759*, 2026.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2019. URL https://nscl.csail.mit.edu/.

Jiayuan Mao, Tomás Lozano-Pérez, Joshua B. Tenenbaum, and Leslie Pack Kaelbling. PDSketch: Integrated domain programming, learning, and planning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36972–36984, 2022.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.15182.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e6d58fc68c0f3c36ae6e0e64478a69c0-Abstract-Conference.html.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Jianing Qi, Jiawei Liu, Hao Tang, and Zhigang Zhu. Beyond semantics: Rediscovering spatial awareness in vision-language models. *arXiv preprint arXiv:2503.17349*, 2025.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2403.04253.

Ashish Seth, Utkarsh Tyagi, Ramaneswaran Selvakumar, Nishit Anand, Sonal Kumar, Sreyan Ghosh, Ramani Duraiswami, Chirag Agarwal, and Dinesh Manocha. EGOILLUSION: Benchmarking hallucinations in egocentric video understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28461–28480, 2025. URL https://aclanthology.org/2025.emnlp-main.1446/.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.19236.

Tom Silver, Rohan Chitnis, Nishanth Kumar, Willie McClinton, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Joshua B. Tenenbaum. Predicate invention for bilevel planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 12120–12129, 2023. doi: 10.1609/aaai.v37i10.26429.

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Scaling up memory for robotic control via experience retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1dH4ARGdwD.

Moritz Tenorth and Michael Beetz. KnowRob: Knowledge processing for autonomous personal robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4261–4266, 2009. doi: 10.1109/IROS.2009.5354602. URL https://ai.uni-bremen.de/papers/tenorth09knowledge.pdf.

Marcel Torne, Andy Tang, Yuejiang Liu, and Chelsea Finn. Learning long-context diffusion policies via past-token prediction. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1744–1755. PMLR, 2025. URL https://proceedings.mlr.press/v305/villasevil25a.html.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World models for physical robot learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240. PMLR, 2023.

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12156–12163. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801581.

Yiqing Xu, Jiayuan Mao, Yilun Du, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and David Hsu. Set It Up!: Functional object arrangement with compositional generative models. In *Robotics: Science and Systems*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.095.

Yiqing Xu, Linfeng Li, Cunjun Yu, and David Hsu. "Stack It Up!": 3D stable structure generation from 2D hand-drawn sketch. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 124–151. PMLR, 2025.

Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, Wengang Zhou, Yao Mu, and Tai Wang. EventVLA: Event-driven visual evidence memory for long-horizon vision-language-action policies. *arXiv preprint arXiv:2606.20092*, 2026.

Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R. Walter. Statler: State-maintaining language models for embodied reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15083–15091, 2024. doi: 10.1109/ICRA57147.2024.10610634.

Juntu Zhao, Wenbo Lu, Di Zhang, Yufeng Liu, Yushen Liang, Tianluo Zhang, Yifeng Cao, Junyuan Xie, Yingdong Hu, Shengjie Wang, Junliang Guo, Dequan Wang, and Yang Gao. Do you need proprioceptive states in visuomotor policies? *arXiv preprint arXiv:2509.18644*, 2025.