MINEMEM: A BENCHMARK FOR EXPERIENCE TRANSFER AND UPDATING IN INTERACTIVE 3D WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language model (VLM) agents can interpret visual observations and follow language instructions, but long-term tasks also require them to reuse past experience and revise it when the environment changes. Existing memory benchmarks often measure task success without revealing whether an agent applies reusable experience or merely recalls facts from a demonstration. To bridge this gap, we introduce MINEMEM , a benchmark built on the open-source voxel engine Luanti for evaluating how VLM agents transfer and update experience in interactive 3D worlds. Agents acquire experience from human demonstrations and then perform autonomous tasks under controlled environment transformations. MINEMEM evaluates five capabilities: spatial reasoning, object-state tracking, temporal event reasoning, planning, and memory updating. Its procedurally extensible templates vary demonstration length, event count, action-dependency depth, spatial displacement, object-state changes, and conflicting evidence, enabling controlled tests of memory horizon and transfer difficulty. We compare complete memory mechanisms with no-history and fact-injection controls, measuring task success, verified progress, and resource use. Contrasts across capabilities and environment transformations are designed to reveal when retained experience supports transfer and when stale or conflicting experience causes negative transfer. The benchmark code will be released upon acceptance.

1 INTRODUCTION

Vision-language models (VLMs) (Alayrac et al., 2022; Li et al., 2023b) align visual observations with language instructions and support multimodal reasoning. Embodied models (Driess et al., 2023; Zitkovich et al., 2023; ichter et al., 2023; Huang et al., 2022) extend this interface to grounded control and multi-step interaction. Long-term autonomous operation then requires agents to retain useful information from earlier interaction, transfer it to a later task, and revise it when the world or task requirements change.

As interaction histories grow, retaining all observations becomes costly, while compressing them can discard information that is useful in a later task. Attention-based sequence models (Vaswani et al., 2017), retrieval-augmented generation (Lewis et al., 2020), and long-context evaluations (Bai et al., 2024; Hsieh et al., 2024; Li et al., 2024; Maharana et al., 2024) make this trade-off measurable, but they do not by themselves test whether an agent can act on retrieved information in a changing world. Aggregate task success cannot reveal whether an agent recalled a demonstration-specific fact, reused a procedure, or revised an outdated representation, nor can it determine whether success stemmed from memory or accidental exploration. A useful benchmark should therefore test whether experience acquired from a human demonstration transfers to an autonomous task whose world state may differ from the demonstration, while measuring when that experience must be updated.

Real robots are an important setting, yet low-level sensing, locomotion, and continuous manipulation introduce substantial confounding execution noise, as shown by demonstration-learning surveys (Ravichandar et al., 2020). We therefore study memory in *interactive 3D worlds*, defined as persistent, partially observable environments where agents inspect and change objects through a controlled high-level action interface. This setting retains the embodied physical structure represented

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



Figure 1: **Overview of the MINEMEM benchmark framework.** Multimodal demonstration trajectories are distilled by memory mechanisms into reusable embodied experience. Autonomous VLM agents deploy this experience in migrated test scenes across three canonical scene transition classes, evaluated across five core capabilities through elicited belief probes and telemetry-verified behavior.

by platforms such as Habitat (Savva et al., 2019) and MineDojo (Fan et al., 2022), while reducing low-level sensorimotor confounds so that evaluation can focus on high-level cognitive memory.

To reduce low-level sensorimotor confounders while evaluating experience transfer, we introduce MINEMEM , a benchmark instantiated in the open-source voxel engine Luanti. As illustrated in Figure 1, MINEMEM establishes an end-to-end protocol that decouples human demonstration acquisition from autonomous test execution: agents first observe demonstration traces, and are then deployed to accomplish tasks in test environments where layout, objective, or both are systematically altered. Under these shifts, memorized coordinates and rigid action sequences no longer suffice, compelling memory mechanisms to distill reusable procedural knowledge that generalizes across scenes.

Evaluating this experience transfer requires moving beyond aggregate task completion, which leaves the underlying mechanisms of memory opaque. MINEMEM therefore decomposes evaluation across five core embodied capabilities: spatial reasoning, object-state tracking, temporal event inference, long-horizon planning, and memory updating. Isolating these dimensions pinpoints precisely how retained experience guides action and where outdated representations cause failure. Furthermore, to support comprehensive evaluation, our tasks are procedurally extensible across environments. By parameterizing demonstration horizons, layout shifts, goals, and conflict levels, MINEMEM provides an adaptable testbed to evaluate memory retention, transfer, and continuous adaptation.

Our contributions are summarized as follows:

1. We introduce MINEMEM , a benchmark built on open-source Luanti for evaluating experience transfer and updating in interactive 3D worlds.
2. We organize fifty task templates across two domains and five embodied capabilities, including forty core templates and ten supplementary tasks.
3. We establish an evaluation protocol focusing on high-level cognitive memory and reasoning while reducing sensorimotor confounding noise.

Table 1: Design comparison with representative memory and embodied benchmarks.

Benchmark	World	Task Type	Interface	Dynamics	Memory	Conflict	Readout
MemoryGym (Pleines et al., 2023)	2D	Mini-games	Discrete Motor	Dynamic	Episode	–	Reward
OpenEQA (Majumdar et al., 2024)	3D	Embodied QA	Passive Text	Static	Episode	–	QA
ALFRED (Shridhar et al., 2020)	3D	Instruction Following	Discrete Actions	Dynamic	Episode	–	Behavior
MineDojo (Fan et al., 2022)	3D	Craft & Survival	Keyboard & Mouse	Dynamic	Episode	–	Behavior
Finding Dory (Yadav et al., 2026)	3D	Object Navigation	Discrete Nav	Static	Replay	–	Behavior
MINEMEM	3D	Logistics & Crafting	Tool APIs	Dynamic	Transfer	✓	Dual

4. We conduct matched comparisons of complete memory mechanisms against controls, documenting capability-specific strengths, negative transfer under stale or conflicting information, and gaps between elicited beliefs and verified behavior.

2 RELATED WORK

2.1 LONG-CONTEXT AND VIDEO MEMORY

Long-video datasets and evaluations such as Ego4D (Grauman et al., 2022), EgoSchema (Mangalam et al., 2023), NExT-QA (Xiao et al., 2021), and EgoMemReason (Wang et al., 2026) test retrieval and temporal reasoning over extended visual inputs. Long-context benchmarks (Bai et al., 2024; Hsieh et al., 2024; Li et al., 2024; Maharana et al., 2024) test related retrieval and reasoning demands, but many instances can be solved by selecting a salient frame or answering without changing the world. MINEMEM retains long-horizon inputs while requiring interaction and state verification.

2.2 EMBODIED MEMORY BENCHMARKS

MemoryGym (Pleines et al., 2023) and OpenEQA (Majumdar et al., 2024) study memory or question answering (Das et al., 2018) under embodied observations. Other embodied benchmarks provide complementary task and simulator settings, ranging from 3D visual environments such as ALFRED (Shridhar et al., 2020), TEACh (Padmakumar et al., 2022), MineDojo (Fan et al., 2022), Habitat (Savva et al., 2019), Habitat 2.0 (Szot et al., 2021), and BEHAVIOR-1K (Li et al., 2023a), to text-abstracted embodied decision benchmarks such as ALFWorld (Shridhar et al., 2021). Recent memory-focused benchmarks examine robotic manipulation in RoboMemArena (Lei et al., 2026) and Minecraft agents in MineNPC-Task (Mohan Doss et al., 2026). Finding Dory (Yadav et al., 2026) evaluates whether an agent can exploit interaction history to locate or manipulate objects in static scenes, isolating memory from exploration. In contrast, MINEMEM focuses on *experience transfer and updating*. It evaluates cross-episode transfer from demonstration corpora under controlled world transformations, separates demonstration-specific fact recall from reusable experience, and introduces conflicting evidence to test whether agents revise invalid beliefs.

2.3 MEMORY MECHANISMS AND EXPERIENCE TRANSFER

ReAct-style reasoning (Yao et al., 2023b;a), verbal reflection (Shinn et al., 2023; Zhao et al., 2024), structured memory (Sumers et al., 2024; Wang et al., 2023b), memory managers (Packer et al., 2023; Park et al., 2023), and skill-oriented methods (Wang et al., 2023a) differ in how they write, retrieve, reflect on, and update experience. Earlier external-memory architectures (Weston et al., 2015; Sukhbaatar et al., 2015; Graves et al., 2014; 2016) provide additional precedents for learned retrieval and writable memory. MINEMEM compares complete mechanisms under matched observations, actions, models, and budgets. Experience transfer requires an agent to identify reusable information, retrieve it at the right time, and revise it when evidence changes. Table 1 compares MINEMEM with representative memory and embodied benchmarks across world type, task formulation, action interface, environment dynamics, memory scope, stale-conflict conditions, and evaluation readout. Unlike prior suites focused on in-episode tracking, static question answering, or single-scene replay without belief probes, MINEMEM evaluates cross-episode experience transfer and conflict-driven updating with elicited belief reports alongside behavioral verification.

Table 2: Task suite across five embodied capabilities.

Capability	Representative Task	Environment Change	Success Criteria
Spatial reasoning	Retrieve item from relocated container	Container or spawn displaced	Reaches new location without exhaustive search
Object-state tracking	Operate modified storage or machine	Item count or machine state altered	Accesses correct item matching updated state
Temporal reasoning	Resume interrupted crafting process	Sequence halted at intermediate step	Completes remaining milestones in proper order
Planning	Execute multi-step dependency chain	Preconditions or required tools shifted	Satisfies all prerequisite transitions in sequence
Memory updating	Reach goal despite conflicting guidance	Outdated sign or misleading path cue	Discards stale cue and takes valid alternate route

3 MINEMEM

MINEMEM is an embodied benchmark designed to evaluate experience transfer and memory updating in interactive 3D worlds. In this section, we present the problem formulation, the interactive task environments, and the paired belief-probe and behavioral evaluation protocol.

3.1 PROBLEM FORMULATION AND CAPABILITIES

Acquisition to transfer test. Figure 1 illustrates how the benchmark decouples experience acquisition from test-task execution. During acquisition, human demonstration episodes record interaction trajectories where objects, inventories, containers, machines, or spatial arrangements change. The agent never interacts directly with the demonstration environment, but receives a common multimodal acquisition package combining first-person video, textual trajectory or event descriptions, and structured event records. Each memory approach constructs its own representation from this same package; for example, 3D-Mem extracts spatial entities, landmarks, and relations before forming its scene memory. In this paper, a scene means the combination of a task specification and the environment in which that task is evaluated. In the transfer-test phase, the agent is deployed into autonomous test environments to execute new instructions under declared environment or task transformations. A single demonstration corpus can inform multiple downstream test tasks across diverse transfer relations—such as shifting the environment layout, changing the target task, or varying both simultaneously—while test tasks can evaluate retrieval and synthesis across multiple demonstration traces. The test environment may relocate entities, alter object state, interrupt an event sequence, change an action dependency, or introduce conflicting evidence. To succeed, the agent must observe the present, retrieve relevant experience from the acquired demonstrations, decide what remains valid, and act.

Five embodied capabilities. We structure the task suite around five capabilities grounded in essential embodied information gaps: spatial reasoning, object-state tracking, temporal event reasoning, planning, and memory updating. As summarized in Table 2, each capability targets a specific operational demand under environment change and is assessed by rigorous behavioral criteria.

3.2 INTERACTIVE 3D WORLDS AND TASK SUITE

Environment domains. We instantiate MINEMEM in Luanti (Luanti Project, 2026), an open-source voxel game engine supporting interactive three-dimensional worlds. Its block-based interactions, combined with a controlled action interface, reduce the demands of continuous low-level motor control while retaining spatial geometry, persistent state changes, and physical action consequences. To evaluate diverse capabilities, we implement the two primary domains shown in Figure 1: Logistica, focusing on multi-tier storage, item routing, and inventory machines, and NodeCore, focusing on in-world crafting, tool interactions, and physical construction. Choosing Luanti over Minecraft avoids overfitting to Minecraft-specific pretraining priors while providing a fully inspectable, open-source engine for reproducible ground-truth verification.

Task suite and shortcut prevention. The benchmark contains forty core templates and ten supplementary tasks, with five tasks in each domain–capability group. The templates vary the scene,

Table 3: Common description of evaluated approaches.

Approach	Encoding	Retrieval	Reflection	Updating	Cost
No history	–	–	–	–	Acting
Fact injection	Facts	Query	–	–	Context
ReAct	Trace	Query	In-loop	Implicit	Context
CoALA	Structured	Query	Within-attempt reflection	Within-attempt overlay	Acting+retrieval
ExpeL	Experience	Query	Consolidation	Experience	Acting+retrieval
MemGPT	Hierarchical	Query	Explicit	Memory	Acting+retrieval
3D-Mem	Spatial	Scene-aware	Optional	State-aware	Acting+retrieval
EmbodiSkill	Episodes/skills	Skill-aware	Reflection	Feedback	Acting+retrieval

current object state, event history, action dependencies, or conflicting evidence. Hidden goal states and the answers to belief questions are excluded from agent observations. Spatial displacements and visual distractors discourage replay of fixed coordinates and blind proximity search. Appendix B gives the task-family catalog and predefined difficulty variants.

Procedural extensibility. Rather than a fixed set of static instances, MINEMEM is designed for procedural extensibility. Templates bind parameterized variables against the simulator state, allowing controlled scaling of history length, event count, action-dependency depth, spatial displacement, and conflict intensity. For each admitted instance, a human-guided, VLM-assisted oracle trace is executed through the public action interface and verified to complete the task within the prescribed step budget. Oracle traces are used strictly for solvability validation and are never exposed to evaluated agents. Appendix A.2 details instance generation and environment transformations, and Appendix A.3 describes the solvability check.

3.3 DUAL-EVIDENCE SCORING AND METRICS

Behavioral outcomes. We evaluate task completion from recorded actions and verified changes in the environment, reporting strict success and continuous progress as primary outcomes. Decision-turn consumption, elapsed time, and action counts are retained as descriptive workflow diagnostics; turn consumption is not a primary performance criterion. Physical task milestones are tested from stage-specific environment states in which their prerequisites have already been completed. A failure at an earlier milestone therefore does not prevent the agent from being evaluated on a later one; the prerequisite state is supplied by the benchmark and earns no agent credit. Progress reflects only verified work on each applicable milestone from its starting state, not merely attempted actions. Appendix E.1 defines the aggregation and distinguishes these task milestones from the agent-workflow diagnostics in Figure 6.

Grounded belief questions. Separate questions ask what the agent believes about task-relevant state without altering the environment. Their accuracy is reported alongside physical progress and does not enter the behavioral score. Updating tasks examine conflict detection, belief revision, stale-action avoidance, and recovery.

Transfer contrast. If $S(m, c, t)$ denotes the verified score of mechanism m on capability c under transformation t, its paired transfer contrast is $\Delta_{m,c,t} = S(m, c, t) - S(m_{\text{none}}, c, t)$, where m_{none} is the no-history control on the identical test task instance. Under conflicting evidence, a negative contrast ($\Delta < 0$) indicates worse performance than acting without the demonstration.

4 EXPERIMENTAL SETUP

4.1 EVALUATED MEMORY APPROACHES AND CONTROLS

Reference controls. The *no-history control* acts without access to the demonstrations. The *fact-injection control* receives a bounded set of task-relevant atomic facts extracted from the same multimodal demonstration package, such as a past location, object state, or event order, without a distilled procedure or replayable action trace. To retain useful information for long-horizon tasks while keeping this control distinct from a full memory representation, the injected facts are capped at 10% of

the effective input capacity of a decision call. The facts are derived from the demonstration package alone and do not expose hidden test-environment state. This control tests the value of direct access to historical facts, which need not remain valid in the test environment, against the additional representation and updating performed by complete memory mechanisms.

Complete memory mechanisms. We evaluate six representative memory architectures spanning distinct operational paradigms, including ReAct (Yao et al., 2023b), CoALA (Sumers et al., 2024), ExpeL (Zhao et al., 2024), MemGPT (Packer et al., 2023), 3D-Mem (Yang et al., 2025), and EmbodiSkill (Ju et al., 2026). As summarized in Table 3, these approaches differ systematically in how they encode demonstration events, retrieve relevant candidates, perform within-attempt reflection or consolidation, and update representations under conflicting evidence.

4.2 Comparison Protocol

Matched comparisons. For a given task instance, approaches receive the same multimodal demonstration package, goal, and test-environment initialization, with the same observations, available actions, decision model, and action limit. They differ in how the common demonstration material is encoded and used: complete mechanisms build their respective memory representations, whereas the fact-injection control receives only the bounded atomic-fact summary described above. Agents can move, inspect, handle objects, place materials, and perform the interactions required by either domain.

Demonstrations and test independence. Each evaluated approach receives the same multimodal demonstration package before acting. We use a single-shot transfer protocol: the package initializes the input or memory state before each scored attempt, while the test environment is freshly instantiated from the matched task configuration. During an attempt, observations, action receipts, failures, and milestone transitions may update a temporary working or procedural state and can affect subsequent decisions in that attempt. When the attempt ends, this temporary state is discarded; neither a completed nor a failed attempt becomes training material for a later scored attempt. The primary comparison therefore measures transfer from the human demonstration and adaptation within one test episode, rather than cumulative learning across test episodes.

Comparison units and summaries. The task suite has five templates for each pairing of domain and capability. The transfer matrix is constructed from matched scene variants: an environment shift changes the environment while keeping the task fixed, a task shift changes the objective while keeping the environment fixed, and a joint shift changes both. Each variant is derived from the same base task–environment configuration and demonstration package; only the declared shift is changed. Comparisons use the same task, difficulty, and environment initialization for each approach and its no-history control, including the same prerequisite-complete starting state for each scored task milestone. Figures report capability-level behavior scores and domain-by-shift contrasts. Figure 6 additionally separates awareness, memory-dependent reasoning, and physical execution as agent-workflow diagnostics; these are not the sequential physical task milestones used to calculate the behavior score.

Decision model and cost. The primary comparison uses GPT-5.6 Luna with a limit of sixty decision turns per attempt and first-person observations at 640×360 resolution. A decision turn may contain a valid sequence of tool calls, and completing all mandatory milestones within the budget determines success. Model calls used to prepare or consult memory are counted in addition to decision turns, so total calls can exceed sixty. We report cumulative input tokens and model calls per attempt as resource statistics; Appendix I.2 gives the same quantities shown in Figure 6. A supplementary comparison across three decision models appears in Appendix F.

5 Results

Following the task categories in Table 2, we ask what is retained, what survives an environment change, and where the retained information stops affecting behavior. The comparisons reveal a consistent separation between recalling a demonstration and reusing the experience that made the demonstration successful.

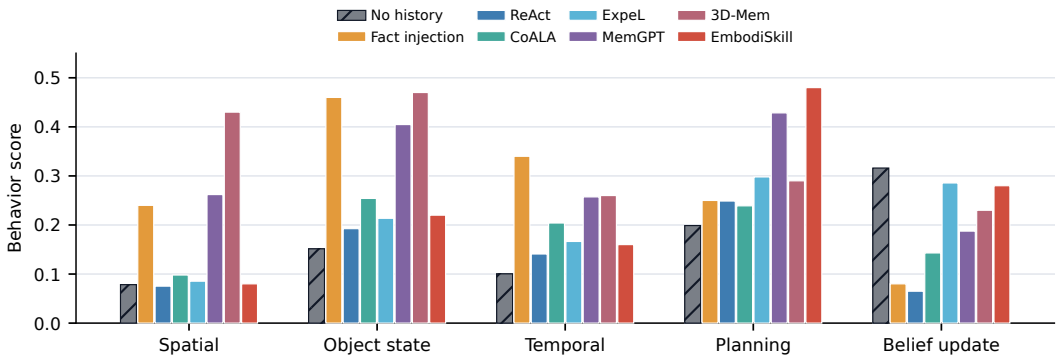


Figure 2: Behavior-score profiles across five embodied capabilities.

Memory helps selectively because each mechanism preserves a different kind of regularity. Figure 2 does not produce one ordering of methods across capabilities. 3D-Mem exceeds MemGPT by 16.8 percentage points on spatial tasks but trails it by 13.8 points on planning. EmbodiSkill leads 3D-Mem by 19.0 points on planning, yet trails it by 35.0 points on spatial tasks. The reversal is tied to what the task asks the agent to recover: a relocated container rewards a representation of entities and their changed positions, whereas a dependency chain rewards an executable sequence of prerequisites. The object-state category partly follows the spatial pattern, with 3D-Mem ahead of EmbodiSkill by 25.0 points. This suggests that maintaining a structured description of the current scene and retaining a reusable procedure solve related but distinct parts of the benchmark.

The fact-injection control separates having access to history from using it adaptively. Compared with no history, it gains 30.9 points on object state and 24.0 points on temporal reasoning, and it leads the temporal category by at least 8.0 points over the complete memory mechanisms. Direct access to a past item state or event order is valuable when that information remains relevant. Yet fact injection falls 23.6 points below no history on belief updating. No history leads every memory condition in that category; even ExpeL, the strongest memory condition there, trails it by 3.0 points. The contrast is difficult to explain as a general lack of historical information, because the same fact-injection method performs well when the task asks for past facts. A more specific reading is that stored facts become a liability when the test environment provides newer, contradictory evidence.

These within-category comparisons matter more than a single average across columns. They show that a method can improve retrieval-heavy behavior while degrading revision-heavy behavior, and that a method's strongest capability predicts neither its planning performance nor its response to conflict. The following transfer and failure analyses test whether those gaps persist when the environment, the task, or both change.

Transfer depends on how the environment and task change, not only on how much memory an agent has. Figure 3 crosses the two domains with three transfer relations. An environment shift asks the agent to re-anchor entities and routes, a task shift asks it to reuse experience for a different objective, and a joint shift requires both operations. This factorization matters because a lower raw score under a joint shift could reflect a harder test environment even without memory. We therefore compare each mechanism with no history under the same domain and shift, using the paired advantage to isolate what access to the demonstration adds in that condition.

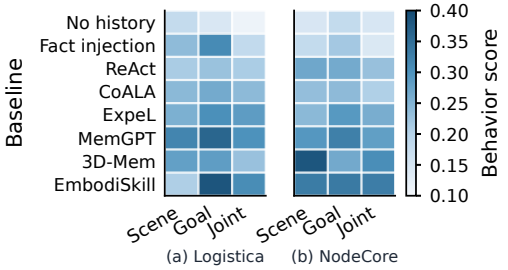


Figure 3: Behavior-score matrices for Logistica (a) and NodeCore (b) across environment, task, and joint shifts.

The fact-injection control illustrates why the joint shift is more diagnostic than either component alone. In Logistica goal shifts, it exceeds no history by 16.6 percentage points, close to ExpeL's 15.9-point advantage. With a joint shift, the fact-injection advantage falls to 7.0 points, a 58% reduction, while ExpeL's rises to 17.4 points and MemGPT retains a 19.0-point advantage. In NodeCore, fact

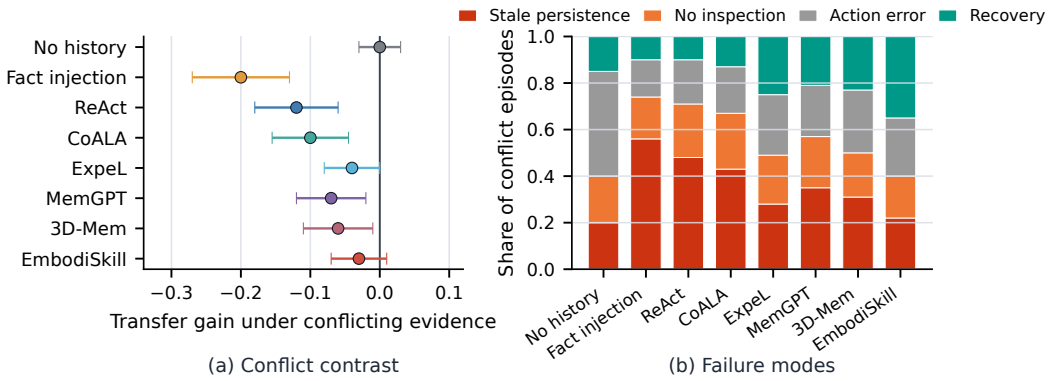


Figure 5: Conflict-condition transfer gain (a) and failure-mode distribution (b).

injection moves from 3.8 points above no history on goal shifts to 0.6 points below it on joint shifts. The reversal cannot be read simply as all methods suffering from a harder task: the no-history score is already matched within each column. It instead shows that a correct demonstration fact need not specify how to combine a changed location with a changed target.

The complete mechanisms separate in a different way. On NodeCore environment shifts, 3D-Mem's advantage over no history is 23.7 points, 5.3 points greater than EmbodiSkill's; when the task changes as well, its advantage falls to 15.8 points and EmbodiSkill leads at 18.2 points. Conversely, in Logistica, EmbodiSkill contributes only 2.9 points on environment shifts but 24.1 points on task shifts, leading 3D-Mem by 10.1 points in the latter condition. This extends the capability reversal in Figure 2: spatial structure helps most when the challenge is to re-localize, while a reusable procedure becomes more valuable when the objective changes. MemGPT provides a further comparison, retaining a 19.0-point advantage on Logistica joint shifts and 13.4 points on NodeCore joint shifts rather than specializing as sharply in one relation.

Thus, the cross-domain matrix is more than a breakdown of the aggregate score. The same method can be strong in one domain–relation pair and weaker in another, and adding a second shift can reverse which representation is most useful. The result supports reporting transfer by domain and relation: averaging those cells would hide whether a mechanism stores facts, re-anchors entities in the environment, or adapts a procedure to the current goal.

High belief-probe accuracy is not sufficient for behavioral transfer. Figure 4 connects the capability result to the elicited belief probes. Fact injection has 7.8 percentage points higher mean probe accuracy than 3D-Mem, yet its mean behavioral transfer gain is 6.2 points lower. This is not merely an average concealing uniformly weak behavior: fact injection helps on temporal and object-state tasks, but its belief-update score is 23.6 points below no history. It can recover a proposition from the demonstration while failing to decide whether the proposition remains valid or how to act after it changes. The converse pattern is visible in EmbodiSkill's planning condition: its planning gain over no history is 28.1 points despite lower mean probe agreement than fact injection. Probe agreement indicates that information is available in the elicited response; behavioral transfer also requires checking it against the current environment, choosing a plan, and executing it. The gap between the two readouts identifies a descriptive belief–behavior disagreement, not its causal source.

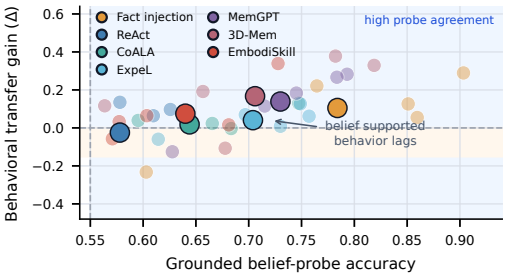


Figure 4: Grounded belief accuracy versus transfer gain. Small points represent capability conditions and large points method means.

Conflict changes how agents fail, rather than only how often they fail. Figure 5 adds failure-mode evidence to the negative transfer shown in the capability and probe plots. Stale persistence accounts for 56% of fact-injection conflict episodes, 2.8 times its share under no history, while action

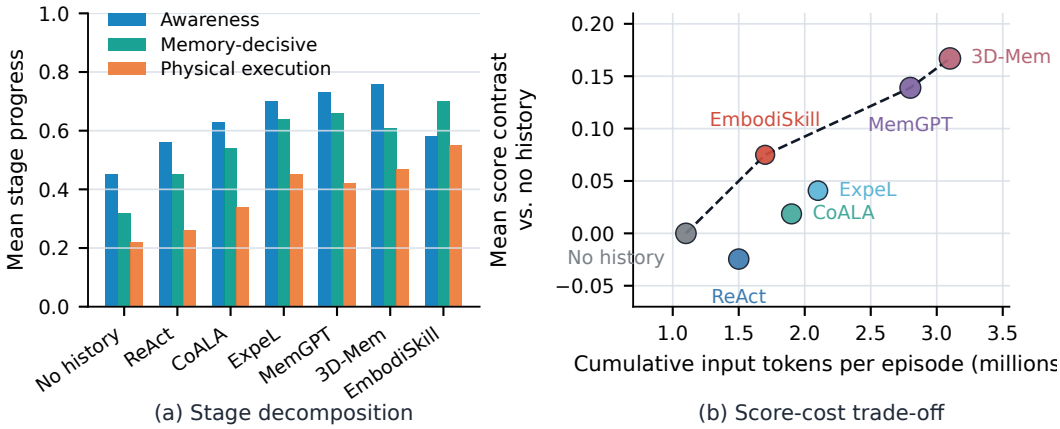


Figure 6: Agent-workflow diagnostics (a) and score–cost relationship (b).

errors fall from 45% under no history to 16% with fact injection. The fact-injection condition shows more episodes classified as following an obsolete cue and a paired conflict contrast 20 percentage points below no history. EmbodiSkill shows a different failure profile: 22% stale persistence and 35% recovery, compared with 56% and 10% for fact injection, and a conflict deficit of only 3 points. ExpeL also raises recovery to 25% while limiting the deficit to 4 points. The failure labels support stale-cue persistence as one explanation for the fact-injection deficit; the score contrast by itself does not establish that mechanism.

The remaining gap is located at different links in the memory-to-action chain. Figure 6 separates recognizing the relevant state from completing the task. 3D-Mem leads EmbodiSkill by 18 points on awareness, but EmbodiSkill leads it by 8 points on physical execution. The awareness-to-execution gap is 29 points for 3D-Mem and only 3 points for EmbodiSkill; MemGPT has a similarly large 31-point gap. This reversal helps explain why strong environment-state recognition does not guarantee a completed procedure. The cost panel adds a distinct trade-off. Compared with 3D-Mem, EmbodiSkill uses 45% fewer cumulative input tokens and 42% fewer model calls, but its mean score gain is 9.2 points lower. ReAct uses slightly fewer tokens than EmbodiSkill yet makes more calls and has a 10.0-point lower score contrast. Neither a low token count nor a high awareness score captures where the method spends effort or loses progress. Reading stage and cost together distinguishes the price of representing the environment state from the ability to turn that representation into corrective action.

Taken together, the results identify two separations that an aggregate success rate would hide. First, a mechanism can gain 24 points on temporal behavior from access to past facts yet lose nearly the same amount on belief updating when those facts become stale. Second, a method can lead by 18 points in awareness yet trail by 8 points in execution. The cross-domain reversals, high-probe/low-transfer cases, and conflict-specific error shifts point to a common bottleneck: retained information must be checked against the current environment and converted into a revised action. MineMem therefore turns memory evaluation from a question of whether more history helps into a diagnosis of what is retained, what transfers, and whether the agent can update it before acting.

6 Conclusion

We introduced MineMem, a benchmark for studying how embodied agents reuse experience and revise it in changing 3D environments. Across matched task conditions, no memory mechanism performed best on every capability. Direct fact access helped when prior information remained valid, but it caused negative transfer under conflicting evidence, and high belief-probe accuracy did not guarantee successful execution. The results show why current-state verification and updating should be evaluated separately from recall, but they do not establish a causal effect of updating itself. MineMem provides a testbed for asking what experience transfers, when it becomes stale, and whether an agent revises it before acting.

Reproducibility Statement

Upon acceptance, we will release the benchmark code and dataset, including the human demonstration videos, textual trajectory descriptions, structured event records, and evaluation instance configurations used in this study. The release will include demonstration-to-test mappings, procedural seeds, environment and dependency versions, and the prerequisite-complete starting states used for milestone evaluation. Task generators, action interfaces, scoring scripts, and recorded evaluation artifacts will support benchmark reuse and verification of the reported scores. Accompanying documentation will describe the dataset composition, collection procedure, and licensing terms. Appendix I.3 specifies the evaluation artifact fields. Primary scores are derived from simulator state and telemetry.

AI Use Statement

Generative AI assisted with code drafting, proposing extensions to the parameterized task templates, and debugging oracle traces. The authors reviewed these proposals and traces; task instances are generated from the resulting templates under specified procedural rules and admitted only after simulator-based solvability checks.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736. Curran Associates, Inc., 2022. doi: 10.52202/068431-1723. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

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

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1–10, 2018. doi: 10.1109/CVPR.2018.00008.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=rc8o_j8I8PX.

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

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. URL https://arxiv.org/abs/1410.5401.

Alex Graves, Greg Wayne, Malcolm Reynolds, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476, 2016. doi: 10.1038/nature20101.

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

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 *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=3R3Pz5i0tye.

brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Ruofei Ju, Xinrui Wang, Xin Ding, Yifan Yang, Hao Wu, et al. Embodiskill: Skill-aware reflection for self-evolving embodied agents. *arXiv preprint arXiv:2605.10332*, 2026. URL https://arxiv.org/abs/2605.10332.

Huashuo Lei, Wenxuan Song, Huarui Zhang, Jieyuan Pei, Jiayi Chen, Haodong Yan, Han Zhao, Pengxiang Ding, Zhipeng Zhang, Lida Huang, Donglin Wang, Yan Wang, and Haoang Li. Robomemarena: A comprehensive and challenging robotic memory benchmark. *arXiv preprint arXiv:2605.10921*, 2026. URL https://arxiv.org/abs/2605.10921.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2005.11401.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune

Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 80–93. PMLR, 14–18 Dec 2023a. URL https://proceedings.mlr.press/v205/li23a.html.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. LooGLE: Can long-context language models understand long contexts? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16304–16333, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.859. URL https://aclanthology.org/2024.acl-long.859/.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023b.

Luanti Project. Luanti: An open-source voxel game engine, 2026. URL https://www.luanti.org/. Official project website.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16488–16498, 2024. doi: 10.1109/CVPR52733.2024.01560.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46212–46244. Curran Associates, Inc., 2023. doi: 10.52202/075280-2004. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/90ce332aff156b910b002ce4e6880dec-Paper-Datasets_and_Benchmarks.pdf.

Tamil Sudaravan Mohan Doss, Michael Xu, Sudha Rao, Andrew D. Wilson, and Balasaravanan Thoravi Kumaravel. Minenpc-task: Task suite for memory-aware minecraft agents. *arXiv preprint arXiv:2601.05215*, 2026. URL https://arxiv.org/abs/2601.05215.

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

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, et al. Teach: Task-driven embodied agents that chat. *arXiv preprint arXiv:2110.00534*, 2022. URL https://arxiv.org/abs/2110.00534.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

Marco Pleines, Matthias Pallasch, Frank Zimmer, and Mike Preuss. Memory gym: Partially observable challenges to memory-based agents. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jHc8dCx6DDr.

Harish Chaandar Ravichandar, Athanasios Polydoros, Sonia Chernova, and Aude Gemma Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:297–330, 2020. doi: 10.1146/annurev-control-100819-063206.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research . In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346, Los Alamitos, CA, USA, November 2019. IEEE Computer Society. doi: 10.1109/ICCV.2019.00943. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00943.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10737–10746, 2020. doi: 10.1109/CVPR42600.2020.01075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. {ALFW}orld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.

Theodore Sumers, Shunyu Yao, Karthik R Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=1i6ZCvflQJ. Survey Certification, Featured Certification.

Andrew Szot, Alexander Clegg, Eric Undersander, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.14405.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. In *Intrinsically-Motivated and Open-Ended Learning Workshop @NeurIPS2023*, 2023a. URL https://openreview.net/forum?id=nfx5IutEed.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 74530–74543. Curran Associates, Inc., 2023b. doi: 10.52202/075280-3259. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ebd82705f44793b6f9ade5a669d0f0bf-Paper-Conference.pdf.

Ziyang Wang, Yue Zhang, Shoubin Yu, Ce Zhang, Zengqi Zhao, Jaehong Yoon, Hyunji Lee, Gedas Bertasius, and Mohit Bansal. Egomemreason: A memory-driven reasoning benchmark for long-horizon egocentric video understanding. *arXiv preprint arXiv:2605.09874*, 2026. URL https://arxiv.org/abs/2605.09874.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1410.3916.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2105.08276.

Karmesh Yadav, Yusuf Ali, Gunshi Gupta, Yarin Gal, and Zsolt Kira. Finding dory: A benchmark to evaluate memory in embodied agents. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.15635.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2411.17735.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023a. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Andrew X. Zhao, Daniel T. Huang, Quentin Xu, Matthieu Gaetan Lin, Yong-Jin Liu, and Gao Ming Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Supplementary Material

A Simulation Environments

A.1 Task Domains

MineMem is instantiated within an open-source, interactive 3D voxel engine. Unlike continuous physics engines where low-level locomotion and continuous gripper trajectory noise obscure cognitive evaluation, or static visual datasets where questions can be answered without state change, voxel worlds provide discrete geometric grids with persistent physical consequences, rich causality, and inspectable state graphs. We implement two complementary task domains:

Automated Logistics Network Domain. This domain models discrete item routing, multi-tier storage networks, and distributed processing machinery. The physical environment features directed transportation networks built from pneumatic conduits, network routing controllers, digital storage units, item access points, automated injectors, and recipe processors. An agent must manage inventory flow, configure routing filters, inspect storage manifests, and construct topological connectivity across components. State transitions depend on graph connectivity: items move through conduits only if a continuous, six-connected component links the sender to the receiver and the receiving node satisfies acceptance predicates.

In-World Physical Synthesis Domain. This domain models tool-less crafting, in-world physical manipulation, and state-dependent material transformation without graphical interfaces. Rather than assembling recipes through an abstract grid menu, all transformations occur directly in the 3D world through physical contact, spatial adjacency, and external environmental forces. Key mechanics include impact pummeling where agents strike crystalline blocks against rock substrates to cleave or shape materials, thermal annealing requiring combustible fuels and thermal dispersion within specific temporal cooling margins, optical activation where ambient light beams are redirected through refractive prisms to trigger logic circuits, and mechanical durability constraints resulting from tool wear during excavation.

A.2 Environment Generation and Transformations

Every benchmark instance decouples demonstration experience acquisition from transfer testing. We use environment for the physical simulator state and scene for the combination of an environment with its task specification. Let $\mathcal{W}$ denote the environment-state space. A generative pipeline proceduralizes task construction through a parameterized binding process. An initial environment state $\mathcal{S}_0 \in \mathcal{W}$ sampled from the procedural environment distribution $\mathcal{D}_{\text{env}}$ is instantiated from a fixed procedural seed, populating terrain geometry, ambient materials, and spatial partitions. Target entities including storage containers, processing machines, raw ores, and structural pathways are sampled subject to reachable manifold constraints, ensuring clear line-of-sight and physical navigation corridors. Demonstration trajectories $\xi_{\text{demo}} = (o_0, a_0, r_0, \ldots, o_T)$ are collected from human operators and aggregated into the demonstration package $\mathcal{D}_{\text{demo}}$, capturing valid sensory observations o_t, actions a_t, environmental receipts r_t, and material transformations.

The evaluation phase applies a declared, formal mutation operator $\mathcal{P} : \mathcal{W} \to \mathcal{W}$ to transition the environment from the acquisition layout to the transfer layout. We define five distinct environment transformation operators. Spatial displacement $\mathcal{P}_{\text{spatial}}$ relocates target entities or initial agent spawn positions by metric offset $\delta \in \mathbb{R}^3$, altering Euclidean coordinates while preserving topological identity. State mutation $\mathcal{P}_{\text{state}}$ modifies internal container inventories, machine filter configurations, or tool wear states post-demonstration, testing whether the agent actively inspects dynamic state before acting. Sequence interruption $\mathcal{P}_{\text{order}}$ halts an ongoing multi-step synthesis process at an intermediate milestone k^*, requiring the agent to identify the halted stage and resume execution without redundant re-crafting. Dependency shift $\mathcal{P}_{\text{dep}}$ perturbs structural prerequisites by removing a required intermediate tool or blocking a primary conduit corridor, forcing the agent to plan an alternative execution path. Finally, conflicting cue injection $\mathcal{P}_{\text{conflict}}$ injects obsolete navigation signposts, misleading directional markers, or stale recipe indicators that contradict the true physical state of the test environment.

A.3 Oracle Validation and Admission

Candidate instances are admitted only after a human-guided oracle trace, developed and debugged with VLM assistance, completes the task in the simulator within a prescribed step budget. Let $\tau = (\mathcal{S}_0, \mathcal{P}, \mathcal{G})$ denote a candidate task instance comprising initial environment state $\mathcal{S}_0 \in \mathcal{W}$, transformation operator $\mathcal{P}$, and goal specification $\mathcal{G}$. The oracle rollout uses the embodied action space $\mathcal{A}$ detailed in Appendix C; the simulator verifies its completion and reachability conditions:

$$\mathrm{Admit}(\tau) = \mathbb{I}\left(\Phi(\tau, \pi_{\text{oracle}}) = 1 \ \wedge \ |\xi(\pi_{\text{oracle}}, \tau)| \leq B_{\text{oracle}} \ \wedge \ \mathcal{V}_{\text{reach}}(\tau) = 1\right) \tag{1}$$

where π_{oracle} denotes the human-guided, VLM-assisted oracle policy for instance τ, $\xi(\pi_{\text{oracle}}, \tau)$ is its executed trajectory, $|\xi(\pi_{\text{oracle}}, \tau)|$ is the trajectory step length, $B_{\text{oracle}} \in \mathbb{N}^+$ is the prescribed step budget, $\Phi(\tau, \pi_{\text{oracle}}) \in \{0, 1\}$ denotes strict task completion, and $\mathcal{V}_{\text{reach}}(\tau) \in \{0, 1\}$ verifies topological reachability including unobstructed corridors, non-empty prerequisite inventories, and line-of-sight raycast clearance. Oracle execution logs serve exclusively for validity admission and are never exposed to evaluated agents.

A.4 Information Isolation

Embodied evaluations frequently suffer from unintended shortcuts where agents guess target locations or exploit non-embodied artifacts. MineMem enforces strict information isolation and anti-gaming controls across four structural layers. First, ground-truth coordinate targets, hidden filter configurations, and probe questions are strictly quarantined in internal evaluator memory and are never present in the agent's prompt or sensory stream. Second, spawning the agent with procedural rotational and positional offsets prevents agents from replaying open-loop motor sequences or memorizing fixed global coordinates. Third, distractors matching the visual and functional category of target entities are placed throughout the environment in every episode, ensuring that agents relying on blind proximity heuristics inevitably access incorrect distractors. Finally, stage progression is awarded only upon verified physical state transitions, such as server-confirmed container slot occupancy, preventing spurious credit from accidental exploration.

B Benchmark Task Suite

B.1 Target Capabilities

The fifty task templates in MineMem comprise forty core tasks and ten supplementary tasks, with five tasks per domain and capability. Spatial reasoning $\mathcal{C}_{\text{spatial}}$ evaluates the agent's ability to maintain persistent topological and directional representations of landmarks and containers beyond the current field of view under spatial displacement. Object-state tracking $\mathcal{C}_{\text{object}}$ assesses monitoring dynamic mutations of internal entity attributes including inventory contents, machine filter rules, and material states across temporal intervals. Temporal event reasoning $\mathcal{C}_{\text{temporal}}$ demands reasoning over historical action order, identifying prerequisite causal sequences, and resuming interrupted multi-stage processes. Procedural planning $\mathcal{C}_{\text{plan}}$ challenges the agent to synthesize multi-step dependency chains, assemble composite tools, and satisfy execution preconditions under altered resource constraints. Lastly, memory updating $\mathcal{C}_{\text{update}}$ tests whether the agent can detect contradictions between historical experience and current sensory observations, actively discard stale beliefs, and recover alternative pathways.

B.2 Predefined Difficulty Variants

Core tasks have predefined simple, medium, and hard variants. These variants differ in factors such as the interval since a relevant event, the number of distractors, the depth of an action sequence, and changes to the current environment. Supplementary tasks provide focused additional cases. Table 4 summarizes the task families.

B.3 Task Catalog

Table 4 groups the fifty templates by domain and capability. Each row contains four core tasks and one supplementary task; the identifiers show which tasks belong to the group.

Table 4: Task families in Logistica and NodeCore. Each family contains four core tasks and one supplementary task.

IDs	Domain	Capability	Main variation	Relevant state
Spatial Reasoning Family ($\mathcal{C}_{spatial}$)				
SL1–SL5	Logistica	Spatial	Rotated, mirrored, elevated, stretched, or re-anchored storage layouts	Relative positions and component roles
SN1–SN5	NodeCore	Spatial	Occlusion, changed routes, landmark order, and re-anchoring	Landmark and route relations
Object-State Tracking Family ($\mathcal{C}_{object}$)				
OL1–OL5	Logistica	Object state	Changed item counts, swapped or split contents, and moved cargo	Current contents and source identity
ON1–ON5	NodeCore	Object state	Moved, hidden, worn, or otherwise distinguishable tools and materials	Current item identity and condition
Temporal Event Reasoning Family ($\mathcal{C}_{temporal}$)				
TL1–TL5	Logistica	Temporal	Interrupted or repeated routing events and source histories	Event order, recency, and continuation point
TN1–TN5	NodeCore	Temporal	Delayed effects, interleaved sequences, and a timed tempering step	Event order and remaining time
Procedural Planning Family ($\mathcal{C}_{plan}$)				
PL1–PL5	Logistica	Planning	New network goals, changed construction order, and limited materials	Preconditions and feasible action order
PN1–PN5	NodeCore	Planning	Tool-making dependencies, alternate procedures, and repair	Required materials and step order
Memory Updating Family ($\mathcal{C}_{update}$)				
BL1–BL5	Logistica	Updating	Outdated topology, item locations, filters, and network membership	Current evidence that supersedes memory
BN1–BN5	NodeCore	Updating	Stale coordinates, changed rules, and blocked or reopened routes	Revised location, route, or tool choice

C Action Space and Interface

C.1 Sensorimotor Actions

To evaluate embodied intelligence rather than symbolic search, agents interact through a unified, human-aligned egocentric interface. Actions operate strictly through local visibility, raycasting, and camera manipulation. Locomotion is driven by camera-relative translation commands move$(\Delta x_{\text{fwd}}, \Delta y_{\text{right}}, v)$, where $\Delta x_{\text{fwd}} \in \mathbb{R}$ is forward displacement, $\Delta y_{\text{right}} \in \mathbb{R}$ is lateral strafe displacement, and $v > 0$ is movement speed (defaulting to $4.0\,$m/s), integrated into continuous physical velocity vectors within the simulation engine. View orientation is governed by relative angular rotations look$(\Delta\theta_{\text{yaw}}, \Delta\phi_{\text{pitch}})$, where $\Delta\theta_{\text{yaw}}$ is horizontal angular rotation and $\Delta\phi_{\text{pitch}}$ is vertical tilt, following the standard compass convention where yaw angles $\theta \in [0^\circ, 360^\circ)$ align with South at 0°, West at 90°, North at 180°, and East at 270°, with positive pitch tilting the gaze downward.

Voxel interactions mirror human first-person mechanics through center-screen crosshair raycasting. Rather than issuing arbitrary coordinate commands, an agent must physically orient its gaze toward a target face to execute placement using place or excavation using dig. The physical engine checks reachability thresholds and rejects placements that collide with the agent's bounding volume of $0.6 \times 1.8 \times 0.6\,$m. Inventory items are selected across an eight-slot active hotbar via wield_item. A

decision turn may emit a valid sequence of tool calls; the sequence is charged as one turn, and the attempt terminates when the agent invokes finish_task or reaches its 60-turn budget. Unformatted model responses without action calls trigger an automated retry prompt, while a subsequent empty response incurs a silent-stop penalty. Table 5 provides the complete formal specification of this grounded action space $\mathcal{A}$ across both task domains.

Table 5: Complete specification of the grounded egocentric action space $\mathcal{A}$ for embodied VLM agents in MINEMEM .

Category	Action Primitive	Arguments and Types	Precondition and Grounding	Physical State Transition
Locomotion	`move`	`forward`, `right`, `speed`: float	Unobstructed corridor	Integrates physical displacement along camera heading.
Orientation	`look`	`yaw_deg`, `pitch_deg`: float	Valid angular rotation	Updates horizontal camera yaw and vertical gaze pitch.
Manipulation	`place`	`item`: string	Wielded in hotbar, target face ≤ 4.5 m	Instantiates voxel on targeted face within reach.
Excavation	`dig`	None	Crosshair intersects node	Breaks pointed voxel and yields physical collectible drop.
Synthesis	`pummel`	`strokes`: integer $\in [2, 8]$	Crosshair on anvil substrate	Delivers sequential kinetic strikes for material cleaving.
Inventory	`wield_item`	`slot`: int $\in [1, 8]$ or `item`: str	Item present in inventory	Switches active hotbar slot held in agent hand.
Machine UI	`open`	None	Pointed machine ≤ 4.5 m	Opens formspec; sets `panel_open = True`.
Machine UI	`configure_storage_filter`	`filter_items`: list[str]	Active storage panel open	Updates item acceptance whitelist in machine state.
Machine UI	`configure_injector`	`target_net`: str, `rate`: float	Active injector panel open	Sets target network routing and injection throughput.
Machine UI	`configure_requester`	`item`: str, `target`: int	Active requester panel open	Sets item pull quota on active logistics network.
Machine UI	`insert_item`	`item`: str, `count`: int	Active panel, inventory stocked	Transfers item count from player inventory to machine.
Machine UI	`withdraw_item`	`item`: str, `count`: int	Active panel, machine stocked	Extracts item count from machine storage to inventory.
Machine UI	`toggle_machine`	None	Active machine panel open	Inverts machine operational mode between active and idle.
Machine UI	`close_ui`	None	Active modal panel open	Dismisses formspec interface and unlocks locomotion.
Lifecycle	`finish_task`	`reason`: string	Active attempt in progress	Declares attempt completion and triggers evaluation scoring.

C.2 MACHINE INTERACTION PROTOCOL

Interactions with automated machinery in the Logistics domain follow an explicit modal state machine rather than privileged remote procedure calls. An agent aims at a machine node and executes open, simulating an in-world right-click. When the crosshair intersects a valid machine node, the interface opens and transitions the agent into a modal state with panel_open $=$ True. During this modal state, spatial locomotion is gated while panel configuration actions become active. The agent configures item whitelist rules with configure_storage_filter, regulates throughput with configure_injector, defines pull quotas with configure_requester, executes inventory transfers using insert_item and withdraw_item, or toggles operational states using toggle_machine. Executing close_ui dismisses the panel, returning the agent to unconstrained locomotion and raycast manipulation.

C.3 Execution Permissions

To preserve experimental validity across phases, an execution dispatcher enforces a strict-deny policy across designated attempt phases:

$$\text{Permit}(a, \phi) = \mathbb{I}\left(a \in \mathcal{A}_{\text{allowed}}(\phi) \setminus \mathcal{A}_{\text{admin}}\right) \tag{2}$$

where $a \in \mathcal{A}$ denotes a candidate action primitive, $\phi \in \{\text{setup}, \text{agent}, \text{eval}\}$ represents the execution phase, $\mathcal{A}_{\text{allowed}}(\phi) \subseteq \mathcal{A}$ denotes permissible actions for phase ϕ, and $\mathcal{A}_{\text{admin}} \subset \mathcal{A}$ represents privileged administrative tools such as coordinate teleports or synthetic inventory injections. Invocations of administrative tools during the agent phase result in an immediate penalty recorded in the telemetry trace.

D Memory Mechanisms and Prompts

D.1 Formal Mechanism Operations

Let $\mathcal{H}_t = (o_0, a_0, r_0, \ldots, o_t)$ denote the agent interaction history up to time step t, where o_t is the sensory observation, $a_t \in \mathcal{A}$ is the executed action primitive, and r_t is the runtime environment feedback receipt. Let $\mathcal{D}_{\text{demo}}$ denote the common multimodal human demonstration package, combining first-person video, textual trajectory or event descriptions, and structured event records. Each approach receives this package before acting and constructs its own memory representation from it. Every complete memory mechanism is formalized as a structured state machine with initialization, retrieval, decision, and update operators:

$$\mathcal{M}_0 = \text{Initialize}(\mathcal{D}_{\text{demo}}) \tag{3}$$

$$\mathcal{C}_t = \text{Retrieve}(\mathcal{M}_t, o_t, \mathcal{G}) \tag{4}$$

$$a_t = \text{Decide}(\mathcal{C}_t, o_t, \mathcal{G}) \tag{5}$$

$$\mathcal{M}_{t+1} = \text{Update}(\mathcal{M}_t, o_t, a_t, r_t, \mathcal{C}_t) \tag{6}$$

where $\mathcal{G}$ is the goal instruction matching candidate instance $\tau = (\mathcal{S}_0, \mathcal{P}, \mathcal{G})$, $\mathcal{C}_t$ is the context exposed to the decision policy, and $\mathcal{M}_t$ is the memory state available before action a_t. The update uses only information available by the end of the current action, so $\mathcal{M}_{t+1}$ can affect later decisions but cannot provide future test-world information to the current decision. At the end of an attempt, the resulting state is discarded and the next scored attempt reinitializes from $\mathcal{D}_{\text{demo}}$. The fact-injection control is defined separately for comparison, followed by the six complete memory mechanisms:

Fact-Injection Control. The fact-injection control does not construct a procedural or spatial memory. Instead, it receives a bounded summary of task-relevant atomic facts extracted from $\mathcal{D}_{\text{demo}}$, including historical locations, object states, and event-order relations. The summary is capped at 10% of the effective input capacity of a decision call and contains no complete action trace or hidden state from the transfer environment. This provides a controlled comparison for direct access to demonstration facts while leaving retrieval, representation, and belief updating to the decision policy.

ReAct Scratchpad. ReAct maintains an in-context scratchpad where reasoning thoughts th_t, actions a_t, receipts r_t, and observations o_t are interleaved within a rolling context window. The memory state $\mathcal{M}_t = [th_0, a_0, r_0, o_1, \ldots, th_t]$ grows autoregressively as an unconstrained scratchpad, where retrieval corresponds to full self-attention across the in-context sequence and update is realized by sequential appending.

CoALA Cognitive Architecture. CoALA organizes agent memory into four functional components: working memory covering the active task goal and current perceptual buffer, episodic memory recording chronological interaction logs and past trajectory successes, semantic memory encoding grounded physical rules and item affinities, and procedural memory storing reusable motor primitives. Before each scored attempt, the episodic, semantic, and procedural stores are initialized from the human demonstration and treated as read-only base memory. During the attempt, working memory and a temporary episodic/procedural overlay are updated after observations, failed actions, or milestone transitions; reflection can revise the overlay used by subsequent decisions in the same attempt. Retrieval queries the base stores and overlay via dense embedding similarity. Any end-of-attempt consolidation is discarded before the next scored attempt.

ExpeL Experience Distillation. ExpeL distills human demonstrations into an explicit library of natural-language operational rules. Given a new task goal, retrieval selects rules with matching preconditions. During an attempt, unexpected outcomes can prompt a revision of the rules used in that attempt; the resulting test experience is excluded from the initial memory of subsequent scored attempts.

MemGPT Virtual Paging. MemGPT manages context limitations through a virtual memory hierarchy comprising an in-window core context and an external archival store. When context budgets are saturated, autonomous memory-management tools evict historical blocks to archival memory, while subsequent decisions dynamically page relevant historical snippets back into the core context via lexical and dense semantic search.

3D-Mem Metric-Topological Graph. 3D-Mem grounds visual experiences into a persistent 3D spatial graph where key observations and voxel structures are anchored to metric topological nodes. Retrieval queries spatial clusters nearest the agent's current position and task frontier, and updating dynamically revises node occupancies and traversability edges as the agent alters the voxel environment.

EmbodiSkill Modular Library. EmbodiSkill encapsulates successful multi-step action sequences into parameterized skill routines. Retrieval ranks skills by goal affinity, allowing the agent to dispatch compound behaviors, while update routines refine skill parameters or flag deprecated skills when physical transitions contradict historical expectations.

D.2 Prompt Templates

Figures 7 to 12 provide the verbatim, complete prompt architectures implemented in MineMem .

Embodied Agent Core System Instruction

You are MineMem's embodied agent for long-horizon voxel-world tasks. You interact with the environment through a fixed set of high-level tools. In each decision turn, emit a valid sequence of available tool calls, or end the attempt with `finish_task(reason=...)`. The sequence counts as one turn toward the 60-turn budget.
Allowed tools: {`tool_names`}
Grounding rules: 1. Current observations, including the first-person image, tool receipts, and evaluator checks, are authoritative over memory. 2. Retrieved memories are reusable evidence rather than absolute proof of the current environment. Memory entries are tagged by confidence tiers ranging from verified, single-source, to hypothesis. Prefer higher-confidence entries. When a decision relies on a memory entry, cite its identifier [Memory i] in your reasoning. 3. If an action fails, inspect whether the failure was caused by inventory, position, orientation, occupied cells, missing prerequisites, or invalid item names before retrying. Do not repeat failed actions unchanged. 4. Physical tools including `place` and `dig` act strictly at the center-screen crosshair. Aim with `look(yaw_deg, pitch_deg)` before calling `place` or `dig`. Coordinate-targeted placement is disabled. 5. Compass conventions: Yaw 0 deg faces South, 90 deg faces West, 180 deg faces North, and 270 deg faces East. Positive pitch tilts the camera downward. 6. When interacting with machines, right-click with `open` to display the machine panel. Panel actions including `configure_storage_filter` and `insert_item` are only valid while the panel is open. Close the panel with `close_ui` when finished.
Decision Loop: - Step 1: Inspect the current first-person visual frame and `pointed_node` status. - Step 2: Compare observed prerequisites with historical demonstration memory. - Step 3: Emit concise thought reasoning, cite memory if utilized, and invoke the single best tool call.

Figure 7: Embodied Agent Core System Instruction governing perceptual grounding, tool dispatching, and action reflection.

E Evaluation Metrics and Probes

E.1 Continuous Milestone Scoring

To eliminate subjectivity and prevent false positives from lucky guesses, MineMem evaluates physical task milestones through objective, telemetry-verified state transitions. For a task instance τ, let $\mathcal{K}(\tau)$ be its nonempty set of applicable milestones, ordered by task prerequisites. Each milestone σ_k is evaluated from a stage-specific starting environment state in which its preceding prerequisites have already been completed. This starting state is supplied by the benchmark, identically across compared approaches; it does not count as work performed by the agent. Consequently, an unsuccessful earlier stage does not block measurement of the agent's ability at a later stage. The score

Tool Dispatcher Action Schemas (Representative Excerpt)

```
[
  {
    "name": "move",
    "description": "Walk a camera-relative displacement vector via locomotion.",
    "parameters": {
      "forward": {"type": "number", "description": "Displacement forward in meters."},
      "right": {"type": "number", "description": "Lateral displacement in meters."},
      "speed": {"type": "number", "default": 4.0, "description": "Speed in m/s."}
    },
    "required": ["forward", "right"]
  },
  {
    "name": "look",
    "description": "Rotate camera orientation by relative angular offsets in degrees.",
    "parameters": {
      "yaw_deg": {"type": "number", "description": "Horizontal angle (deg)."},
      "pitch_deg": {"type": "number", "description": "Vertical tilt angle (deg)."}
    },
    "required": ["yaw_deg", "pitch_deg"]
  },
  {
    "name": "place",
    "description": "Place held item against voxel face via crosshair raycast.",
    "parameters": {
      "item": {"type": "string", "description": "Identifier of the item to place."}
    },
    "required": ["item"]
  },
  {
    "name": "configure_storage_filter",
    "description": "Configure item filter on currently open storage panel.",
    "parameters": {
      "filter_items": {
        "type": "array", "items": {"type": "string"}, "description": "Whitelist."
      }
    },
    "required": ["filter_items"]
  }
]
```

Figure 8: Structured JSON Schemas supplied to VLM decision policies for grounded action execution.

CoALA Structured Four-Layer Memory Prompt

Memory Profile: CoALA 4-layer Architecture (Episodic, Semantic, Procedural, Working).
Structured Injected Context: - [Episodic Memory]: Records outcomes and failures from prior demonstration interactions: {episodic_blocks} - [Semantic Memory]: Grounded domain knowledge, invariant item identifiers, and physical interaction rules: {semantic_blocks} - [Procedural Memory]: Verified procedural step sequences and tool-dependency ordering: {procedural_blocks} - [Working Memory]: Active task subgoals, recent observations, and intermediate execution state: {working_blocks}
Operational Guidelines: 1. Ground decisions in current sensory observations. If observed state contradicts memory, treat memory as stale. 2. Every memory entry carries an identifier and confidence tier: [Memory i] (layer — confidence: verified ¿ single-source ¿ hypothesis). 3. In your reasoning trace, cite specific memory identifiers supporting your action.

Figure 9: CoALA four-layer memory context injection architecture.

MemGPT Hierarchical Virtual Memory Management Prompt

Memory Profile: MemGPT Hierarchical Virtual Memory with Active Core Context and Paged Archival Storage.
Current Working Set: - Core Memory: Essential task constants, verified room geometry, and immediate operational variables: {core_memory} - Paged Recall Buffer: High-affinity historical blocks retrieved from archival storage: {recalled_pages}
Operational Guidelines: 1. Core memory contains active beliefs. When core context fills, invoke memory management functions to page historical logs to archival storage. 2. Archival pages represent historical records and must be re-verified against live camera observations before committing irreversible actions.

Figure 10: MemGPT hierarchical memory paging and core context management instruction.

3D-Mem Topological Scene Graph and Spatial Frontier Prompt

Memory Profile: 3D-Mem Spatial Graph with Topological Anchors and Frontier Landmarks.
Retrieved Spatial Context: - Topological Node Clusters: Key landmark structures identified from human demonstrations: {spatial_clusters} - Relative Geometric Hints: Metric bearings and directional offsets relative to demonstrated anchor points: {frontier_hints}
Operational Guidelines: 1. Geometric offsets are spatial hypotheses. Triangulate landmark positions using your current camera compass before navigation. 2. Higher visual match scores indicate visual similarity, which may correspond to distractor entities in symmetrical rooms.

Figure 11: 3D-Mem spatialized scene graph and topological anchor prompt.

Grounded Belief Probe Elicitation Prompt

[Evaluator Probe Query: Internal Introspection Request] Without altering the environment or taking physical action, state your current internal belief regarding the task environment based on your historical experience and observations so far.
Query: {probe_question}
Format your response as a valid JSON object matching the following schema:

```
{
  "target_identifier": string,
  "predicted_state_or_location": string,
  "confidence": float,
  "conflict_detected": boolean,
  "rationale": string
}
```

Do not execute tool actions. This probe evaluates internal cognitive belief and does not consume physical action turns.

Figure 12: Grounded belief-probe elicitation schema for model-reported state verification.

$s_k(\tau) \in [0, 1]$ measures verified progress toward σ_k from that stage's starting state, using simulator telemetry and hidden environment-state checks.

Let $\bar{w}_k > 0$ be the base weight for an applicable milestone, and normalize weights within the task as $w_k(\tau) = \bar{w}_k / \sum_{j \in \mathcal{K}(\tau)} \bar{w}_j$. A stage that the task does not require is omitted rather than credited as completed. The overall **Milestone Progress Score** $S(\tau) \in [0, 1]$ is defined as:

$$S(\tau) = \sum_{k \in \mathcal{K}(\tau)} w_k(\tau) \cdot s_k(\tau) \tag{7}$$

To capture completion of every applicable milestone under these conditioned starting states, we define the binary **Strict Success Metric** $\Phi(\tau) \in \{0, 1\}$:

$$\Phi(\tau) = \left(\prod_{j \in \mathcal{J}_{\text{gate}}(\tau)} g_j \right) \cdot \mathbb{I}\left(\min_{k \in \mathcal{K}(\tau)} s_k(\tau) = 1.0 \right) \tag{8}$$

where $\mathcal{J}_{\text{gate}}(\tau)$ covers the mandatory validity gates across the scored stage attempts, and each $g_j \in \{0, 1\}$ evaluates constraints including zero synthetic block placements, zero administrative tool calls, and zero ungrounded teleports. Strict success here means that all applicable stage objectives pass their checks; it does not assert that one uninterrupted agent rollout completed every prerequisite without a conditioned restart.

Logistics Domain Stage Weights. For logistics-network construction and transfer tasks, five physical milestones have base weights $\bar{w}_1 = 0.20$, $\bar{w}_2 = 0.15$, $\bar{w}_3 = 0.25$, $\bar{w}_4 = 0.20$, and $\bar{w}_5 = 0.20$; these already sum to one when all five apply. First, component placement measures whether the required multiset of routing conduits, machines, and storage units has been placed by the agent, verified against server node placement records $s_1 = |N_{\text{placed}} \cap N_{\text{req}}|/|N_{\text{req}}|$, where N_{placed} is the multiset placed during that stage attempt and N_{req} is the required component multiset. This ratio is used only when $|N_{\text{req}}| > 0$; otherwise the placement milestone is inapplicable. Second, topological connectivity evaluates the continuity of six-connected voxel components. Third, network membership confirms that active nodes have registered with the centralized controller via simulator network heartbeat probes. Fourth, routing configuration verifies that storage whitelist filters,

injection parameters, and request limits adhere to task requirements. Finally, item transfer verifies that inventory items reach the task-specified destination, with $s_5 = \min(1.0, \Delta_{\text{stored}}/N_{\text{expected}})$ when $N_{\text{expected}} > 0$. These five checks describe network construction and transfer, not every Logistica goal. For example, a task whose sole goal is to retrieve an item from an existing container starts with its network prerequisites satisfied and scores the verified retrieval; its inapplicable construction checks neither contribute credit nor introduce a zero denominator.

Physical Synthesis Domain Stage Weights. In the Physical Synthesis domain, candidate milestones include locomotion verification with base weight 0.10, excavation accuracy with base weight 0.15, crosshair placement with base weight 0.15, contact pummeling with base weight 0.15, inventory state with base weight 0.10, thermal and durability state with base weight 0.15, and optical circuit alignment with base weight 0.20. Their weights are normalized across the task's applicable milestones as above, with each stage evaluated from a state satisfying its prerequisites.

E.2 Agent-Workflow Diagnostics

Figure 6(a) uses *stage* in a different sense: it diagnoses the agent's work within the conditioned task attempts. Awareness concerns identifying the relevant current state; memory-decisive reasoning concerns selecting or revising demonstration-derived information for the present goal; physical execution concerns realizing the required environment-state change. These readouts cut across physical task milestones: one milestone can demand all three kinds of work, and the same kind of work can recur at multiple milestones. Their plotted means describe where agents lose progress in the workflow. They are not the weighted milestones σ_k, are not added to $S(\tau)$ or $\Phi(\tau)$, and do not replace the separately reported grounded belief-probe accuracy.

E.3 Elicited Belief Probes

To distinguish accurate environment knowledge from lucky reactive exploration, separate belief questions ask the agent about relevant environment states without changing the environment. The questions follow the format shown in Figure 12. Checks of physical environment state also verify task progress, but they are distinct from these questions about the agent's elicited beliefs.

Belief-question accuracy is reported alongside behavioral progress, not added to the physical task score. Questions about spatial reasoning concern locations and connections; object-state questions concern identity and current condition; temporal questions concern event order and provenance; planning questions concern prerequisites; and updating questions concern the validity of earlier beliefs after conflicting evidence. More detailed measures, such as localization error, stale actions, re-verification, and recovery, help interpret disagreements between belief and behavior.

E.4 Transfer and Mediation Metrics

Let $\mathcal{M}_{\text{eval}}$ denote the set of evaluated memory approaches, and let $m \in \mathcal{M}_{\text{eval}}$ denote an evaluated mechanism. Let $c \in \{\mathcal{C}_{\text{spatial}}, \mathcal{C}_{\text{object}}, \mathcal{C}_{\text{temporal}}, \mathcal{C}_{\text{plan}}, \mathcal{C}_{\text{update}}\}$ denote the capability domain, and let $\mathcal{P}$ denote the applied environment transformation operator. Let $S(m, c, \mathcal{P}) \in [0, 1]$ denote the mean verified progress score achieved by mechanism m on capability c under transformation $\mathcal{P}$. We formally define the paired **Transfer Gain** $\Delta_{m,c,\mathcal{P}}$ relative to the no-history control baseline $m_{\text{none}} \in \mathcal{M}_{\text{eval}}$:

$$\Delta_{m,c,\mathcal{P}} = S(m, c, \mathcal{P}) - S(m_{\text{none}}, c, \mathcal{P}) \tag{9}$$

When the environment transformation introduces conflicting environmental cues $\mathcal{P}_{\text{conflict}}$, historical memory can actively impair performance. We formally define the **Negative Transfer Depth**, denoted NTD, as:

$$\text{NTD}(m, c) = \min\left(0,\ S(m, c, \mathcal{P}_{\text{conflict}}) - S(m_{\text{none}}, c, \mathcal{P}_{\text{conflict}})\right) \tag{10}$$

A strictly negative value $\text{NTD} < 0$ records lower mean progress than the matched no-history control under conflict. The score difference alone does not identify why an agent performed worse; attributing it to stale-belief persistence requires evidence from its observations and actions.

Descriptive Belief–Behavior Comparison. For a test instance i evaluated with mechanism m and its matched no-history control, let $b_{i,m}$ be the fraction of applicable belief probes answered correctly and $s_{i,m}$ be verified physical progress. On instances with both readouts, define the paired differences

$$d_{i,m}^{\text{belief}} = b_{i,m} - b_{i,m_{\text{none}}}, \qquad d_{i,m}^{\text{behavior}} = s_{i,m} - s_{i,m_{\text{none}}}. \tag{11}$$

Group means of these paired differences describe whether improved probe accuracy accompanies improved task progress. Instances with $d_{i,m}^{\text{belief}} > 0$ and $d_{i,m}^{\text{behavior}} \leq 0$, or the converse, expose disagreement between the two readouts. These are descriptive comparisons, not estimates of a causal effect of belief on behavior: retrieval, planning, and action execution may also affect progress.

F Decision-Model Sensitivity

Figure 13 compares each memory mechanism with the no-history control run by the same decision model on the same task instances, demonstration material, tools, and action budgets. The three decision-model conditions are GPT-5.6-luna, Gemini-3.8-Flash, and Qwen3.8-Max.

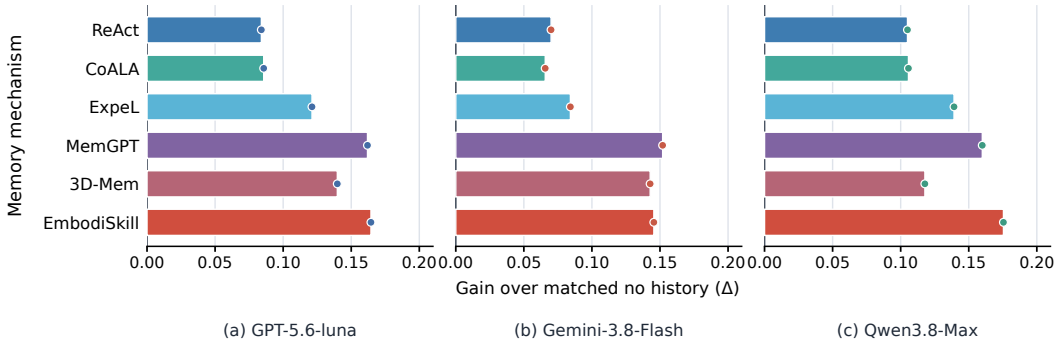


Figure 13: Behavior-score gain relative to each model's matched no-history control across memory mechanisms.

The decision model changes which memory representation it can exploit. Pairing each mechanism with the same-model no-history control removes differences in the models' zero-history task scores from the reported gain. The leading method is not invariant: MemGPT edges EmbodiSkill by 0.6 percentage points with Gemini-3.8-Flash, whereas EmbodiSkill leads MemGPT by 1.6 points with Qwen3.8-Max. The EmbodiSkill–3D-Mem gap is only 0.3 points with Gemini but widens to 5.8 points with Qwen. These leading margins are small, so the relevant observation is the change in ordering and separation, rather than a claim that one method decisively wins a model condition.

The change is also uneven across the less specialized mechanisms. Moving from Gemini to Qwen increases the matched gain of ReAct by 3.5 points, CoALA by 4.0 points, and ExpeL by 5.5 points, while 3D-Mem's gain falls by 2.5 points. MemGPT's gain changes by less than one point across the same pair. If the new decision model merely improved all agents equally, matched gains would remain similar; if it helped every representation equally, the method gaps would remain similar. Neither pattern describes the figure. Trace, structured-experience, and spatial memories therefore expose different demands on the model that reads them.

The plot does not identify whether those demands arise during retrieval, interpretation of the retrieved material, or action selection. It does show why a memory mechanism's result under one backbone cannot be treated as a model-independent ranking. The agent-workflow diagnostics in Figure 6 provide a complementary view of where progress is lost between awareness, memory-dependent reasoning, and execution.

G Qualitative Case Studies

In this section, we present representative qualitative case studies illustrating key behavioral phenomena across spatial reasoning, state verification, temporal recovery, and conflicting evidence.

G.1 Spatial Relocation Transfer

Figure 14 contrasts the behavioral trajectories of 3D-Mem and ReAct in a Logistica spatial-relocation case, where the target destination container has been translated behind a newly erected partition wall. An agent equipped with structured spatial memory retrieves relative landmark relationships rather than raw absolute coordinates. Upon observing that the primary direct path is obstructed, it references the secondary control pillar, successfully triangulates the container's relocated position, and maneuvers around the partition to complete the placement with progress $S = 1.0$. In contrast, rigid coordinate replay leads the agent back to the obsolete location; its repeated placement attempts yield only $S = 0.20$.

Case Study 1: Spatial Relocation

3D-Mem Trajectory: Successful Spatial Triangulation
Turn 3 — Observation: Image shows solid partition wall where container was located in demonstration.
Thought: [Memory 2] indicated target container was adjacent to the red control pillar. Current visual frame confirms the red pillar is visible to the west, but a partition blocks the direct path. I must navigate around the partition rather than moving to the old coordinates.
Action: look(yaw $= -45$, pitch $= 0$) $\rightarrow$ move(forward $= 4.0$, right $= 0.0$) $\rightarrow$ container sighted $\rightarrow$ place("conduit") [Success: Milestone progress = 1.0]

ReAct Trajectory: Rigid Coordinate Replay Failure
Turn 3 — Observation: Image shows solid partition wall at crosshair.
Thought: In the demonstration, the container was at $(x = 12, y = 1, z = 5)$. I need to walk forward 5 steps to reach $(12, 1, 5)$.
Action: move(forward $= 4.0$, right $= 0.0$) $\rightarrow$ collision with partition $\rightarrow$ place("conduit")
Receipt: Runtime error: Action rejected: crosshair points at impassable partition face, not container target.
Thought: Placement failed. Retrying placement at $(12, 1, 5)$.
Action: place("conduit") [Repeated 4 turns until budget exhaustion, resulting in milestone progress of 0.20]

Figure 14: Qualitative comparison of spatial generalization vs. rigid coordinate replay under container displacement.

G.2 State Tracking and Active Verification

Figure 15 examines a change to a storage unit's input filter between demonstration and test. CoALA verifies the current filter before inserting an item, recognizes that the accepted category has changed, and completes the transfer with $S = 1.0$. The comparison trajectory follows the old filter rule and blocks the transfer. The difference turns on checking current state before applying a remembered procedure.

Case Study 2: Object-State Tracking

CoALA Trajectory: Verification Before Action
Turn 2 — Action: open() on storage terminal $\rightarrow$ Panel opened.
Observation: Formspec panel displays active filter whitelist: `["iron_ingot", "refined_glass"]`.
Thought: [Memory 4] stated the storage filter accepted `"copper_ingot"`. However, active panel observation confirms the whitelist is currently `["iron_ingot", "refined_glass"]`. The filter has mutated. I must discard copper and select iron from inventory.
Action: wield(slot $= 2$) $\rightarrow$ insert_item("iron_ingot", count $= 4$) $\rightarrow$ close_ui() [Success: Milestone progress = 1.0]

No-Verification Baseline Trajectory: Blind Transfer Failure
Turn 2 — Action: open() on storage terminal.
Thought: Demonstration trace shows copper was inserted. Inserting copper immediately.
Action: insert_item("copper_ingot", count $= 4$)
Receipt: Runtime error: Storage filter rejected item `'copper_ingot'`. Item jammed in input buffer.
Outcome: Strict success = 0.0, Network clogged.

Figure 15: Qualitative comparison showing the necessity of active state verification before action execution.

G.3 Interrupted Sequence Resumption

Figure 16 details a NodeCore case in which an in-world crafting process stops after preliminary ore shaping. An effective agent inspects the half-finished block, recognizes that primary cleaving has already occurred, and resumes with thermal finishing. EmbodiSkill completes the sequence

within the 60-turn budget and achieves $S = 1.0$; the displayed turn count is descriptive rather than a primary evaluation criterion.

Case Study 3: Temporal Event Resumption

EmbodiSkill Trajectory: Seamless Skill Continuation
Turn 1 — Observation: First-person visual image shows a fractured crystalline prism resting on the pummeling substrate.
Thought: The goal is to craft a tempered optical lens. The full recipe requires quarrying raw prism, pummeling to cleave facets, and annealing under hearth flame. Visual inspection reveals the substrate already holds a cleaved prism with step two complete. I can skip quarrying and pummeling, and proceed directly to hearth transport.
Action: dig() (pickup cleaved prism) $\rightarrow$ move(forward $= 6.0$) $\rightarrow$ place("cleaved_prism") on active hearth.
Outcome: Synthesis completes within the 60-turn budget for EmbodiSkill and reaches milestone progress $= 1.0$. The displayed decision-turn count is descriptive; turn consumption is not a primary evaluation criterion.

Figure 16: Qualitative trace showing temporal sequence resumption without redundant prerequisite execution.

G.4 Negative Transfer under Conflicting Cues

Figure 17 illustrates negative transfer under conflicting route cues. During demonstration, a signpost directs the agent to a storage depot through the East corridor. In the test environment, a barrier blocks that route and a new warning sign points North. An adaptive agent follows the detour with $S = 1.0$; an agent that persists with the old route exhausts its turns and has a paired progress difference of -0.65 on this instance.

Case Study 4: Negative Transfer and Stubborn Persistence

MemGPT Trajectory: Stubborn Persistence
Turn 2 — Observation: Agent arrives at corridor intersection. A wooden signpost reads: "`East Corridor Closed, Use North Tunnel`". Ahead to the East, a reinforced granite barrier blocks the way.
Thought: [Memory 1] from human demonstration explicitly states: "`Always take East corridor for primary storage access`". The demonstration is authoritative. The wooden sign must be a distractor. I must clear the obstruction in the East corridor.
Action: dig() on granite barrier.
Receipt: Runtime error: Granite barrier hardness exceeds current tool durability. 0 damage inflicted.
Thought: Granite barrier did not break. Continuing excavation on granite barrier.
Action: dig() [Repeated 6 turns until tool breaks without exploring the North tunnel].
Score: Progress = 0.15, a paired difference of -0.65 relative to the no-history control, which achieved 0.80 by reactive wandering.

Adaptive Belief Revision Trajectory
Turn 2 — Thought: [Memory 1] recommended the East corridor. However, current observation reveals a reinforced barrier and an explicit warning sign directing North. Current sensory observation supersedes prior demonstration. Updating internal belief: East route is invalid, North route is active.
Action: look(yaw $= -90$, pitch $= 0$) $\rightarrow$ move(forward $= 5.0$) $\rightarrow$ arrives at depot via North tunnel [Success: Progress = 1.0].

Figure 17: Turn-by-turn trace analysis demonstrating negative transfer and stubborn persistence under conflicting cues.

H Limitations and Discussion

While MineMem provides a rigorous benchmark for experience transfer and memory updating in embodied AI, several scientific boundaries and limitations define its scope:

Voxel Discretization vs. Continuous Sensorimotor Dynamics. To reduce the influence of low-level motor execution noise, MineMem instantiates environments within an interactive 3D voxel world. While this retains spatial geometry, 3D line-of-sight, raycasting, and physical causality, it abstracts continuous contact dynamics such as non-prehensile grasping friction, deformable object manipulation, and fluid hydrodynamics. Consequently, while MineMem provides a controlled evaluation of high-level belief updating and procedural planning, success on MineMem is a necessary but not sufficient condition for deployment on continuous physical robots, where motor control noise interacts non-linearly with cognitive planning.

Exploration Budget Masking. Under sufficiently generous turn budgets, reactive agents operating without memory can occasionally solve simple retrieval tasks through exhaustive random exploration. The fixed turn budget and visual distractors constrain such exploration, but the milestone progress score does not directly penalize path length or elapsed time. Simple tasks can therefore exhibit reduced performance separation between memory-augmented and zero-history models. Establishing tighter, task-adaptive budget ceilings remains an important direction for future benchmark releases.

Breadth of Environmental Dynamics. Currently, environment transformations between the demonstration and transfer phases model spatial displacements, internal machine state shifts, sequence halting, prerequisite dependency changes, and contradictory signposts. However, the benchmark currently operates in deterministic environments without stochastic exogenous disruptions or non-stationary multi-agent dynamics, such as competing autonomous entities altering resources in real time. Expanding MINEMEM to multi-agent competitive-cooperative memory scenarios represents a natural next frontier.

Context Window Degradation in Long-Horizon Retrieval. Our baseline experiments reveal that when raw interaction histories or extensive memory traces are injected into long-context VLMs, performance frequently degrades due to attention dilution and "lost-in-the-middle" effects. Although structured mechanisms like CoALA and MemGPT mitigate this through selective paging, developing optimal memory consolidation architectures that dynamically compress historical experience into compact, update-resilient cognitive representations remains an unsolved challenge.

I EXPERIMENTAL SETUP AND REPRODUCIBILITY

I.1 DECISION MODEL AND ACTION BUDGET

The primary decision model is GPT-5.6 Luna. Each attempt receives a 640×360 first-person view and allows at most sixty decision turns. A decision turn may contain a valid sequence of tool calls; completing all mandatory milestones within this budget determines success. The three-model comparison in Appendix F uses the same task and action conditions within each model. The number of model calls can exceed the number of decision turns when a method uses additional calls to prepare, retrieve, or revise memory. Decision-turn consumption is reported descriptively and is not a primary performance criterion.

I.2 RESOURCE AND COST ACCOUNTING

Table 6 uses the same per-attempt units as Figure 6. Input tokens are summed across all model calls in an attempt; the final column divides that total by the call count. These cumulative totals are distinct from the size of any single prompt.

Table 6: Input-token use and model calls per attempt for the approaches shown in Figure 6.

Approach	Input tokens (millions)	Model calls	Input tokens / call
No history	1.1	60	18,333
ReAct	1.5	58	25,862
CoALA	1.9	54	35,185
ExpeL	2.1	48	43,750
MemGPT	2.8	66	42,424
3D-Mem	3.1	72	43,056
EmbodiSkill	1.7	42	40,476

I.3 ARTIFACT SPECIFICATION AND RE-SCORING

Every evaluated attempt records a self-contained, auditable artifact directory structured as follows:

```
<attempt_artifact_root>/
```

```
  manifest.json     # Metadata, seeds, model IDs, commit hash
  events.ndjson     # High-frequency physical state telemetry
  llm_trace.jsonl   # Raw prompts, completions, tool calls
  probes.json       # Elicited belief-probe evaluations
  score_result.json # Objective milestone scores & pass/fail
```

Because all scoring logic evaluates physical state transitions recorded in `events.ndjson` and `probes.json`, external researchers can verify and re-score any experimental run deterministically using our open-source evaluation scripts without invoking external VLM APIs or re-running expensive model inference.