# MEMJEPA: ERASURE-TRAINED MEMORY LETS LATENT WORLD-MODEL PLANNERS SEE THROUGH OCCLUSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A latent world model can imagine the consequences of its actions in a learned state space, enabling model predictive control in robotics. However, an agent often loses sight of what it controls, such as when its arm moves out of the camera's frame. A windowed joint-embedding predictive architecture (JEPA) planner is then blind to the hidden actor, and adding memory trained only on visible clips does not help. We introduce Memory-Augmented JEPA (MemJEPA), which carries a recurrent belief through the planner's imagination and trains it with actor erasure: the actor is removed from the input clips but kept in the prediction targets. Theoretically, we show that a fixed-window latent planner cannot react to a hidden actor's motion that leaves no trace in its window, making memory of earlier frames necessary, and that erasure training against clean targets provides a learning signal for a useful carried state. Empirically, MemJEPA was tested on three image-based control tasks (reaching, pushing, navigation) with the frozen encoder of LeWorldModel (LeWM). MemJEPA raises average success under actor occlusion from 13% to $\sim$75% against a memoryless twin trained on the same data with the same actor erasure, while matching or exceeding its performance with the actor in view. The hidden-view gains transfer, at a smaller size, to a DINO-WM backbone.

## 1 INTRODUCTION

Joint-embedding predictive architecture (JEPA) world models such as LeWorldModel (LeWM; Maes et al., 2026) plan in an image encoder's latent space. They imagine the outcome of candidate actions and pick those that land closest to the goal. Every planner in this lineage (DINO-WM, LeWM, AdaJEPA, Sub-JEPA, FIRM-WM, LEAP and MotionJEPA; Zhou et al., 2025; Maes et al., 2026; Wang et al., 2026; Zhao et al., 2026; Wu et al., 2026; Pham & Bera, 2026; Karmann et al., 2026) is evaluated with the controlled entity in view and predicts from only its last few frames. Once the entity has been out of view for longer than that, the planner no longer knows where it is, and success falls sharply (Figure 1).

Prior methods have tried to add memory, either a recurrent state (Hafner et al., 2019) or a belief fed into the predictor (Zhou & Zhang, 2026; Wu et al., 2026). Trained on ordinary clips, where the entity is always visible, it does not help. A recurrent state-space model (RSSM) plans well with the entity in view, but with the entity hidden it does no better than a planner with no memory.

The missing piece is the training signal. In ordinary training clips the entity is always visible, so the model can read it from the current frame and the state has no reason to remember it. We erase the entity from each training clip's input frames but keep the clean next latents as targets. The targets still depend on the entity, so the model must carry it from earlier frames.

*We find that erasure training is what lets a latent-space planner make use of its memory.* Erasure without a carried state (a memoryless twin trained with the same erasure recipe) and a carried state without erasure (the standard RSSM) both remain at the memoryless floor, whereas combining the two restores planning (Figure 1). A last-visible cache fails once the entity moves, and a context window once the entity has been hidden longer than the window. Swapping the carried state between



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.

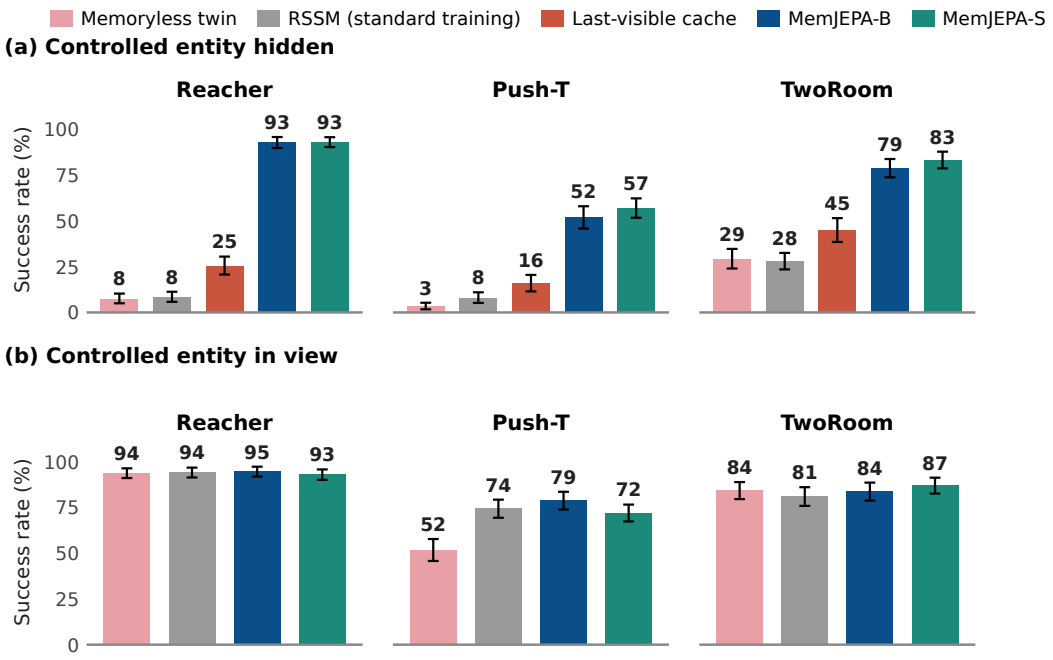



Figure 1: **Success with the controlled entity (a) hidden and (b) in view** (held-out sets; 200 cases $\times$ 3 seeds per bar; 95% case-clustered intervals). All arms share LeWM's encoder and cross-entropy-method (CEM) planner; The memoryless twin uses the same erasure recipe (MemJEPA-B additionally uses an imagination-consistent term). The RSSM is a recurrent world model on the same latents, trained the standard way (Table 4). The cache plans from the last visible frame, which wins when the hidden entity is parked and loses when it moves (Section 4.1). The cache is omitted from (b), since with the entity visible it is the memoryless planner. On Push-T, erasure training lowers the memoryless twin's in-view success (52 against 74 for the standard RSSM), yet the standard RSSM, strong in view, still falls to 8 with the pusher hidden. The entity is hidden from step 10 onward (step 15 on TwoRoom), 10 steps before the first decision. Longer hidden spans are in Figure 4.

paired episodes returns success to the floor. A ridge probe reads the hidden fingertip from the state, not the frame latent.

MemJEPA-B keeps LeWM's windowed transformer predictor and reads a 0.89M-parameter gated recurrent unit (GRU) belief at its input. MemJEPA-S replaces the predictor with the RSSM of Hafner et al. (2019), trained with erasure. Both meet two conditions, erasure training (C1) and a recurrent state advanced inside every imagined candidate of the cross-entropy-method (CEM) planner (C2).

We make five contributions.

1. **Hidden actors break planning.** With the entity hidden, the memoryless twin falls to 7.5, 3.3 and 29.3% success on Reacher, PushT and TwoRoom, against 93.8, 51.7 and 84.3% in full view (Figure 1, Table 2). Occlusion changes only pixels, and the twin is trained on erased clips, so the drop measures the missing entity. Memory also beats the twin on DINO-WM's encoder (Table 3).
2. **Standard recurrence stays blind.** The standard fix, a recurrent state updated along the model's predicted rollouts (Hafner et al., 2019), trained the usual way on our latents, keeps full-view success but stays at 8.5, 7.8 and 27.8% hidden (Table 4): its training frames always show the entity, so its state has nothing to carry (Section 4.3). A context window helps only while fewer frames have been hidden than it holds, and a last-visible cache only when the entity is parked (Figure 4, Section 4.1).
3. **We introduce MemJEPA, a JEPA planner whose memory is trained to track an entity it cannot see.** It erases the entity from training inputs against clean next-latent targets (C1) and advances its state inside every imagined candidate (C2). MemJEPA-B is a GRU belief read at LeWM's predictor input, and MemJEPA-S is the same RSSM trained under both conditions.

4. **We show that erasure-trained memory restores planning with the entity hidden.** MemJEPA-B reaches 93.0, 52.0 and 79.0% hidden and MemJEPA-S reaches 93.2, 57.0 and 83.3% (Table 2). Neither erasure without a state (the twin) nor a state trained without erasure (the standard RSSM) leaves the floor. A state frozen inside candidates still reaches 57.0% on Reacher, and advancing it adds 36 points (Table 4). Swapping the state between paired episodes returns both designs to the floor (Figure 5c). App. H shows that a fixed window cannot react to actor motion that leaves no trace in it, and that erased inputs with clean targets give the state something to learn.
5. **A reusable test.** We release the protocol, entity masks, recipe-matched twins, held-out manifests and re-tally script, so any latent-space planner can be tested the same way.

## 2 Related work

*JEPA world models for planning.* DINO-WM plans with CEM on frozen DINOv2 features (Zhou et al., 2025), and LeWM trains the encoder and predictor end to end with the SIGReg regularizer (Maes et al., 2026; Balestriero & LeCun, 2025). LEAP instead changes the planner, running L-BFGS through the frozen LeWM (Pham & Bera, 2026), and could be combined with our change to the model. Hierarchical JEPA planners add a subgoal generator or predictor above a frozen world model (Masip et al., 2026; Cheng et al., 2026; Zhang et al., 2026; Caselli et al., 2026) or train it jointly with the model (Barbeau et al., 2026), all under full observability.

*Memory in world models.* PlaNet and Dreamer advance a recurrent state in imagination (Hafner et al., 2019; 2023) and S4WM and R2I use state-space memory from the HiPPO and S4 line (Gu et al., 2020; 2022a; Deng et al., 2023; Samsami et al., 2024), but none of them plans over a frozen encoder. CLWM and FIRM-WM read a belief at the predictor's input (Zhou & Zhang, 2026; Wu et al., 2026). CLWM, however, hides a pushable object and trains its encoder end to end with a contrastive counterfactual objective instead of erasure, and FIRM-WM evaluates in full view. Other designs plan from state inputs (Tang et al., 2026), keep an explicit map (Ieong et al., 2026) or stay at toy scale (Radha & Goktas, 2026). We keep a frozen JEPA encoder, hide the actor itself and pair each design with a matched memoryless twin.

*Occlusion and masked training.* Under occlusion, ELVIS plans in a recurrent state-space model (Du et al., 2026), $\mu$VLA carries a recurrent memory inside a vision-language-action (VLA) policy (Cherepanov et al., 2026), and object permanence has been learned from self-imagined occlusions (Traub et al., 2024) and from tracking-based memory (Huang et al., 2024), whereas we add it to a frozen JEPA planner without tracking or object slots. Observational dropout (Freeman et al., 2019), feature masking (Seo et al., 2022) and object-slot masking in Causal-JEPA (Nam et al., 2026) shape representations or robustness, and Loci-Looped trains through frame blackouts against clean targets (Traub et al., 2024), but none of them trains a memory against clean targets for a pretrained, optionally frozen, predictor to read. Table 1 compares the closest works, and the supplementary material (Section S11) covers the rest.

## 3 Architecture

MemJEPA trains a recurrent state on clips with the actor erased from the input against clean next-latent targets (C1) and advances it inside every imagined CEM candidate (C2). Both designs build on LeWM (Maes et al., 2026), as shown in Figure 2. A ViT-Tiny (vision transformer) encoder $e$ maps a frame to a 192-d latent $z_t = e(o_t)$, and a causal transformer predictor $f$ takes the last three latents and actions and predicts the next latent, $\hat{z}_{t+1} = P\big(f(z_{t-2:t}, a_{t-2:t})\big)$, where $P$ is the

Table 1: The closest works (Zhou & Zhang, 2026; Wu et al., 2026; Nam et al., 2026; Traub et al., 2024) against four properties of our setting (n.s.: not stated; C-JEPA: Causal-JEPA).

| | CLWM | FIRM-WM | C-JEPA | Loci-Looped | MemJEPA |
|---|---|---|---|---|---|
| Frozen JEPA-style encoder | no | no | yes | no | yes |
| State advanced in every candidate | yes | yes | no | n.s. | yes |
| Moving entity hidden mid-episode | partly | n.s. | n.s. | yes | yes |
| Matched memoryless twin | no | partly | n.s. | partly | yes |

output projection. Each action block holds five primitive actions, so the window covers at most 15 environment steps. The belief is a recurrent state inside the predictor that every observed latent writes to and every prediction reads from,

$$h_t = \mathrm{GRU}\big(W_{\mathrm{in}}[\tilde{z}_t;\, E_a(a_t)],\; h_{t-1}\big), \qquad h_{-1} = 0, \tag{1}$$

$$\hat{z}_{t+1} = P\big(f(\tilde{z}_{t-2:t}, a_{t-2:t}) + g \odot W_{\mathrm{out}}\,\mathrm{LN}(h_t)\big), \tag{2}$$

where $\tilde{z}_t$ is the latent of the frame actually observed, masked or not, $E_a$ is the action embedding, $W_{\mathrm{in}}$ and $W_{\mathrm{out}}$ are learned projections, and LN is layer normalization. The scale $g$ is learned per channel and initialized to zero, so training starts exactly at the memoryless model, and after training it is a fixed parameter rather than a per-step switch. Equation 2 reads the belief at the predictor's *output*, whereas MemJEPA-B reads it at the *input*, adding to each window token the read of its own step before the positional embedding,

$$\hat{z}_{t+1} = P\big(f(\tilde{z}_{t-2:t} + r_{t-2:t},\, a_{t-2:t})\big), \qquad r_\tau = g \odot W_{\mathrm{out}}\,\mathrm{LN}(h_\tau), \tag{3}$$

with the same zero-initialized $g$. A *middle* read adds $r$ after the third of six blocks (supplementary material, Section S17).

The state is carried across the episode. In the planner, each candidate's imagined rollout advances its own copy of the state with the imagined latent and the candidate action, $h_\tau^{(k)} = \mathrm{GRU}(W_{\mathrm{in}}[\hat{z}_\tau^{(k)}; E_a(a_\tau^{(k)})], h_{\tau-1}^{(k)})$ from $h_t^{(k)} = h_t$ for candidate $k$, and the copies are then discarded, so scoring never changes the committed belief. This *advanced* read is condition C2. The *frozen* read, which some earlier evaluations used (App. K) and whose cost Section 4.2 measures, holds every imagined step at $h_t$. With zero imagined steps the two reads coincide.

The memory adds 892,884 parameters (7.6% of the 11.7M trainable predictor, 4.95% of the whole 18.0M model), a width that saturates TwoRoom but not PushT on the frozen-predictor path (Table 18). The memoryless twin is the same predictor, recipe, data, encoder and initialization without the memory, so it is not parameter-matched. The parameter-matched control, the belief module with its recurrence severed (state reset to zero at every step, same data, recipe and steps), lands with the memoryless twin on Reacher (Table 12).

In our setting, a recurrent state learns to carry a hidden entity only if some training inputs hide it. On half the training clips we therefore erase the task entity from the inputs after the first $k_{\mathrm{vis}}$ frames and keep the unmasked next latents as targets, with the encoder frozen and the other constants in Table 15. Unlike the masking of Section 2, the erased frames are also what the planner sees at test time. The erasure recipe needs only per-frame masks of the task entity (simulator masks here, an off-the-shelf segmenter on real data) and the clips already used to train the world model, and no new demonstrations. The masks cannot be skipped, since a belief trained only on blackouts bridges a blank Reacher camera but not an erased arm (Table 17). The predictor can stay frozen, as training only the 0.89M belief comes within 2.0 hidden-view points of the fine-tuned belief on all three tasks, with intervals inside $\pm 7$ points (Table 12). Fine-tuning the predictor adds 6.7 full-view points on PushT, and on TwoRoom the frozen path is $-2.8\,[-7.0,\, +1.2]$ against the twin with a full view. A replan takes 277.4, 303.3 and 129.7 ms for the memoryless model, MemJEPA-B and MemJEPA-S (App. J).

The planner is CEM (Rubinstein & Kroese, 2004) with LeWM's settings (Table 15) apart from 10 iterations on Reacher, a disclosed deviation (App. G), and minimizes the squared distance $\|\hat{z}_{t+H}(a) - z_g\|_2^2$ between the imagined latent at the planning horizon $H$ and the goal latent.

MemJEPA-B uses the GRU (Cho et al., 2014) belief of Equation 3, advanced at every imagined step of every CEM candidate, and on half the clips an imagination-consistent term feeds the belief the model's own multi-step imagined latents from a random position onward (Bengio et al., 2015; Shah & Shrestha, 2026; Assran et al., 2025). The success bars (Section 4) were fixed on $\mathrm{B}_0$, the belief without this term, and we added the term only after $\mathrm{B}_0$'s held-out evaluations, so MemJEPA-B's extra 1.8 to 4.3 hidden-view points over $\mathrm{B}_0$ are post-selection. A waypoint add-on that sets goals on a learned latent, as in Hafner et al. (2022), is in App. A, and excluded variants are in the supplementary material (Section S13). MemJEPA-S is the RSSM (Hafner et al., 2019) of the standard-trained baseline in Table 4, trained instead on erased clips against clean next-latent targets, with its state copied and advanced inside each candidate in the same way (Figure 2b).

Table 2: Success (%), entity hidden and in full view, held-out sets (200 cases × 3 seeds; reused sets: Table 13). †Post-hoc comparator on sets already evaluated; ‡evaluated after $B_0$ on each set. Window: a memoryless planner with a 10-frame context. Other baselines: a last-visible cache scores 25.3, 15.8 and 45.0 hidden; the 40-frame window fixed in advance (Reacher only) scores 60.0 hidden and 91.8 full. Bold marks the better design where the MemJEPA-S minus MemJEPA-B interval excludes zero (Section 4.3; contrasts: Table 14).

| | Reacher (arm) | | PushT (pusher) | | TwoRoom (agent) | |
|---|---|---|---|---|---|---|
| Model | hidden | full | hidden | full | hidden | full |
| Memoryless twin | 7.5 | 93.8 | 3.3 | 51.7 | 29.3 | 84.3 |
| 10-frame window† | 74.0 | 94.7 | 32.5 | 74.7 | 78.0 | 82.7 |
| MemJEPA-B‡ | 93.0 | **94.7** | 52.0 | **78.8** | 79.0 | 83.8 |
| MemJEPA-S | 93.2 | 93.0 | 57.0 | 72.0 | **83.3** | **87.0** |

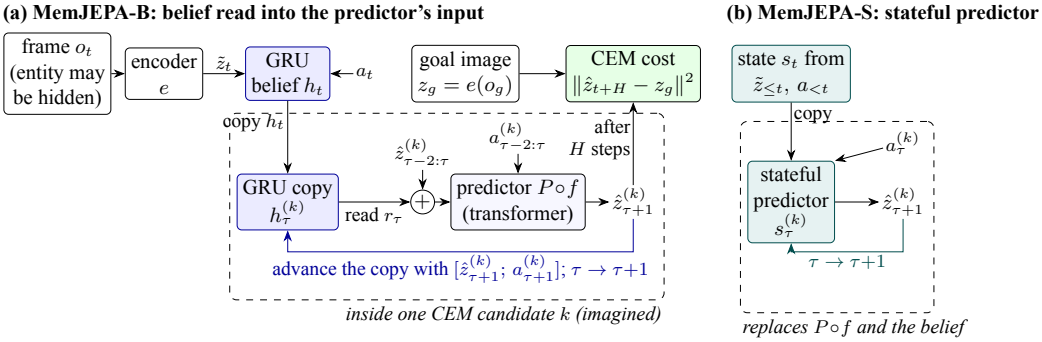



Figure 2: The architecture. (a) MemJEPA-B: a GRU belief $h_t$ is written by every observed latent; inside each CEM candidate a copy is advanced along the imagined rollout and read into the predictor's input tokens, then discarded. (b) MemJEPA-S: an erasure-trained stateful predictor replaces the transformer predictor and the belief; its state is copied and advanced the same way.

## 4 Experiments

We ask (i) where a belief pays off, when it tracks a moving entity or recalls a parked one (Section 4.1), (ii) how it should be trained and read (Section 4.2), (iii) whether erasure training turns a standard recurrent world model into a working memory (Section 4.3), and (iv) whether the gains carry over to a second frozen backbone.

**Protocol.** Each task (Figure 3) has a memoryless floor and a full-view ceiling, and on the same episodes we compare each memory arm with a memoryless twin (recipe-matched for $B_0$ and MemJEPA-S) and with 10- and 40-frame context windows. Each window is a single untuned setting, and on Reacher the 40-frame window scores below the 10-frame one (60.0 against 74.0%). Before any held-out set was evaluated we fixed the case lists, two success bars and two primary contrasts, $B_0$ (MemJEPA-B without its imagination-consistent term, Section 3) and MemJEPA-S, each against its recipe-matched twin, with the cache and, on Reacher, the 40-frame window as comparators. With the entity hidden, the memory arm must beat its twin by at least 20 points (point estimate) with a 95% interval above 0 and beat the cache with an interval above 0, and with a full view its one-sided 95% lower bound against the twin must lie above $-5$. The bars are the same on every task. MemJEPA-B, the 10-frame window, the state swap and other removal controls, the read sites, the advancing ladder and the waypoint head are secondary and were evaluated later on the same sets (Table 13).

**Statistics.** Each held-out set is a case list fixed before training (200 cases × 3 training seeds per cell) and is evaluated once per arm fixed in advance, with later reuses in Table 13. The TwoRoom set had first been evaluated with an earlier own-prediction variant, so its contrasts are second evaluations. Intervals are pair- or case-clustered bootstrap 95% intervals (10,000 resamples, exact McNemar tests where stated) and quantify variation across episodes and cases conditional on the trained models.

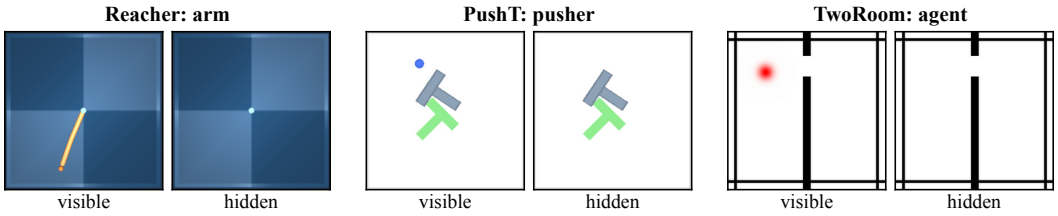



Figure 3: The three tasks with the controlled entity visible and hidden; the goal image never shows it.

Table 3: DINO-WM backbone: difference from the trained memoryless twin (points; held-out sets, 200 cases $\times$ 3 seeds; 95% intervals, hidden only). Details and absolute rates: App. F.

| | Reacher | | PushT | | TwoRoom | |
|---|---|---|---|---|---|---|
| | hidden | full | hidden | full | hidden | full |
| MemJEPA-B | $+13.0\,[+9.0, +17.0]$ | $-5.2$ | $+5.0\,[+1.8, +8.3]$ | $+7.7$ | $+34.2\,[+28.0, +40.2]$ | $+0.0$ |
| MemJEPA-S | $+40.0\,[+35.0, +44.8]$ | $-26.5$ | $+19.3\,[+14.8, +23.8]$ | $-32.3$ | $+43.2\,[+38.0, +48.3]$ | $-15.2$ |

Because the bootstrap pools the three training seeds, these intervals do not estimate variation between training runs, and we treat the seed-level results in the supplementary material (Section S20) as descriptive. Development draws (100 pinned episodes) choose settings and are marked. Tasks, data, evaluators and base checkpoints are the public releases, except three checkpoints we trained to parity (LeWM's PushT and DINO-WM's Reacher and TwoRoom), and the protocol, masks and evaluator are released with the supplement (planner inputs in App. I).

**Tasks and backbones.** We use three tasks, each with the entity hidden and in full view, on two frozen backbones, LeWM's JEPA encoder for the main results and DINO-WM's DINOv2, and on Reacher we also test a 20-frame hidden span. Cube, the fourth LeWM task, failed the 20-point occlusion criterion (Section 4.1), since hiding the cube or the arm costs at most 7 points (App. I). On DINO-WM, when the entity covers under 5% of the encoder's patches, every arm and its twin use an entity-mask-weighted loss ($\lambda = 10$), a rule set after the unweighted PushT run showed that the belief left the prediction loss unchanged. With the entity hidden, both designs beat the trained DINO-WM twin on every task and three of six cells clear the 20-point bar, but only MemJEPA-B passes the full-view bar, on PushT and TwoRoom (Table 3, App. F).

### 4.1 Where the belief pays: tracking versus recall

The occluded LeWM suite consists of three of LeWM's four public tasks, the DeepMind Control Suite's Reacher (Tassa et al., 2018), PushT (Chi et al., 2023) and TwoRoom (Sobal et al., 2025), each with LeWM's release plus an occlusion mask. Each task hides a variable that the goal image does not reveal, and a task is kept only if the memoryless planner loses at least 20 points to its full view (the conditions that failed this test, TwoRoom with the door hidden and Cube, are in App. I).

We test two regimes. In *tracking*, on all three tasks, the entity is erased after the first frames while it keeps moving under the recorded actions, so a cache of the last visible frame is wrong by construction. In *recall*, on occluded PushT, pairs of real episodes share the pusher's start and the goal but differ in the block's pose. The block is erased from step 10 to 84 and planning starts at step 60, so the masked frames of the two members are identical and only each member's own expert continuation succeeds (supplementary material, Section S16). The block does not move while hidden, so here we expect a cache to win.

The three arms (the memoryless predictor, the same predictor planning from a cache of the last visible latent, and the belief inside the predictor) share the encoder, data and erasure recipe, which only the standard-training controls of Table 12 skip, so a gap between the arms is not a recipe effect. The belief here is $B_0$, and MemJEPA-B's imagination-consistent term is shared only by the v2 twin below. The recipe costs the memoryless model 11.8 full-view points on PushT against released

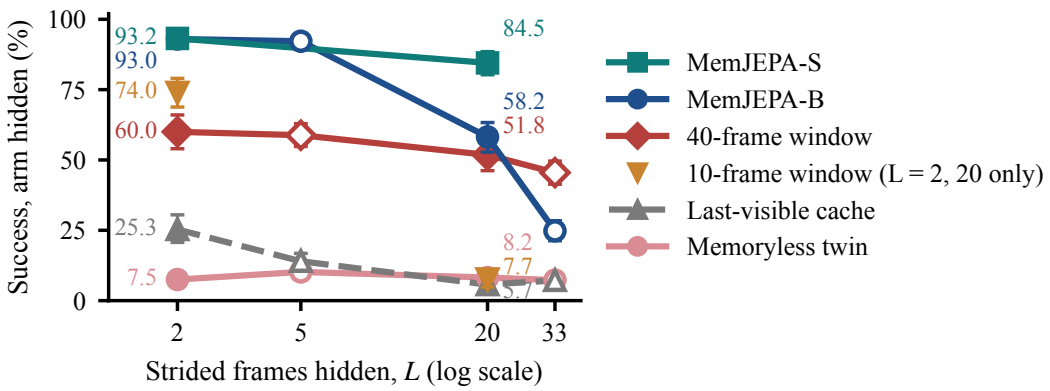



Figure 4: Reacher, arm hidden: success by strided frames hidden (one frame: 5 environment steps; 200 cases × 3 seeds per point; 95% case-clustered intervals). Filled markers: held-out set; open markers: validation set. Contrasts are in the supplementary material, Section S12.

LeWM, which partly explains MemJEPA-B's full-view lead there (goal 50, supplementary material, Section S21). For recall, the PushT models are retrained without any source episode of the 115 pairs.

Success falls with the hidden span, the protocol's one parameter (Figure 4, full-view parity in App. B). With the entity hidden, every belief trained on entity-erased clips and advanced in imagination beats its memoryless twin, both on LeWM (Table 8) and on PLDM (Sobal et al., 2025), a JEPA-style planner with a VICReg-trained convolutional encoder (held-out except where marked, App. E).

**Tracking.** On the held-out sets, $B_0$ beats its paired memoryless twin on every task, by 83.7 points on Reacher (all intervals in Table 8), and beats a last-visible cache by $+67.8\,[+62.0,\ +73.5]$, $+31.8\,[+25.5,\ +38.2]$ and $+30.5\,[+24.5,\ +36.5]$ on Reacher, PushT and TwoRoom, where on Reacher the cache runs on $B_0$'s own model (23.3%). $B_0$ also stays inside the $-5$ full-view bar on every task, so every bar fixed in advance passes. On TwoRoom, MemJEPA-B moreover leads the v2 twin, trained with the imagination-consistent term and the mixed erasure schedule, by $+26.2\,[+18.2,\ +34.0]$ (per-task results in the supplementary material, Section S14). Window blindness is graded. At 2 hidden frames the 10-frame window trails MemJEPA-B by 19.0 points on Reacher, and at 20 hidden frames, past the training clip, the 40-frame window ties MemJEPA-B but not MemJEPA-S, which reaches 84.5% (contrasts in the supplementary material, Section S20). A window can thus help only while fewer frames have been hidden than it holds, and a cache only when the hidden entity does not move (one static construct tested).

**Recall.** On the static recall condition (115 pairs, goal 60 steps ahead), the input-site belief ($B_0$) beats memoryless by 14.6 points, but, as expected, a last-visible cache on the belief's own model (76.8%) beats it by $35.5\,[30.0,\ 40.9]$ points (Table 16 reports 41.3 against 75.2% for the output-site model's cache). The recall set had first been evaluated with output-site arms (Table 13).

### 4.2 Train the belief the way it is read

The main negative result concerns one-step own-prediction training. It lifts the weak output-site belief on the Reacher validation sets, but at the input site it changes success by $-4.7\,[-6.4,\ -3.0]$ points on the Reacher held-out set (five training seeds) and gains nothing on TwoRoom, and the model trained this way fails past the training clip. Planning to look, a planner term that steers toward imagined model disagreement, does not help either (App. H and supplementary material, Section S17).

The belief holds the hidden entity, and the planner uses it (Figure 5, Table 4). A ridge probe decodes the Reacher fingertip from the GRU belief and from MemJEPA-S with $R^2 \geq 0.92$ over 40 hidden steps, against about 0 from the frame latent (supplementary material, Section S12). Advancing the belief with the action alone, which amounts to dead reckoning, reaches only 50.0% (Figure 5b).

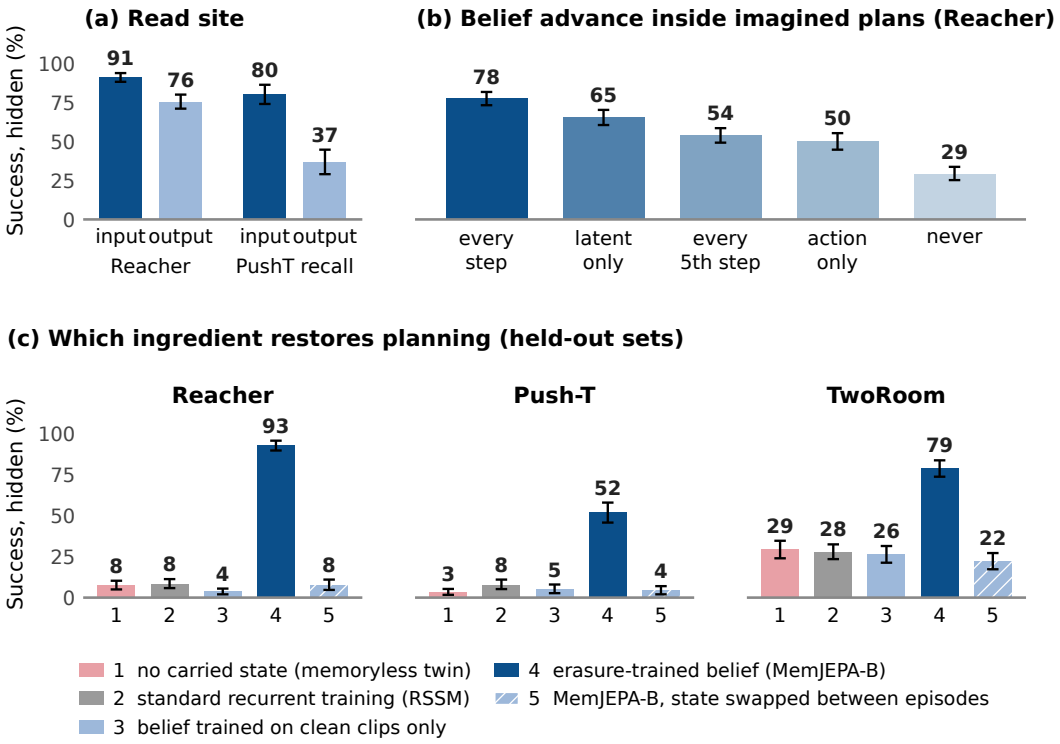



Figure 5: **What makes the belief work** (entity hidden; 3 seeds pooled; 95% case-clustered intervals). (a) Reading the belief at the predictor's input beats reading it at the output. (b) The belief copy works best when advanced with both the imagined latent and action at every step (Reacher confirmation set). (c) Adding one ingredient at a time: only the erasure-trained belief (4) leaves the memoryless floor, and swapping its state (5) sends it back.

**Erasure training, not memory capacity, restores planning.** Figure 5c adds one ingredient at a time. A carried state alone does not help: the standard RSSM (8.5, 7.8 and 27.8%) and a belief trained on clean clips (3.7, 5.2 and 26.3%) stay with the memoryless twin (7.5, 3.3 and 29.3%), although both could remember. On clean clips the entity is visible in every input frame, so the prediction loss is lowest when the model reads it from the current frame, and the state never learns to hold it. Erasure removes that shortcut. The input no longer shows the entity but the target still depends on it, so the only way to lower the loss is to carry the entity forward. The erasure-trained belief then reaches 93.0, 52.0 and 79.0%, and swapping it between paired episodes drops success back to 7.7, 4.3 and 22.2% (MemJEPA-S: 7.2, 2.8 and 21.2%). The gain lives in what the state holds, not in the added parameters or the recipe.

Advancing the state matters on Reacher, where freezing it inside candidates costs MemJEPA-B 36.0 hidden and 18.7 full-view points, 17.3 more hidden than in full view ($-17.3\,[-22.8,\ -11.8]$, post hoc). The frozen model still keeps $+49.5$ over the twin, so C2 adds to a gain that erasure training with a carried state already gives. The effect is confounded with a broken rollout on PushT and not detectable on TwoRoom ($-2.7\,[-7.0,\ +1.7]$), and we did not test why. Proposition 1 (App. H) shows that a window predictor cannot tell apart histories that share its window, and Proposition 2 states what erasure training against clean targets makes the state carry.

### 4.3 MemJEPA-S: an erasure-trained stateful predictor

A 0.9M-parameter stateful predictor (Hafner et al., 2019; 2023) trained the standard way on LeWM's latents stays within 4.5 points of the memoryless twin on every task with the entity hidden (Tables 4 and 7), so in our setting recurrence alone is not enough. Erasure without a carried state (the twin) and a carried state without erasure (this RSSM) both remain at the memoryless floor, and combining them restores planning. MemJEPA-S, the same RSSM trained under both conditions, lifts off the floor on every task (Table 2, intervals in Table 14). With the entity hidden it leads MemJEPA-

Table 4: Conditions and state controls (held-out sets; success %, 200 cases × 3 seeds). Frozen: the state is held fixed inside imagined candidates (post-hoc). No-erasure: evaluated with the entity hidden only (–: not run with the entity in view), contrasted in Table 12 with a re-run erasure-trained partner. The re-run is in Table 7, and intervals are in App. H.

| | Reacher | | PushT | | TwoRoom | |
|---|---|---|---|---|---|---|
| Model | hidden | full | hidden | full | hidden | full |
| Memoryless twin | 7.5 | 93.8 | 3.3 | 51.7 | 29.3 | 84.3 |
| RSSM (Hafner et al., 2019), standard training | 8.5 | 94.2 | 7.8 | 74.3 | 27.8 | 81.0 |
| $B_0$ (no imagination-consistent term) | 91.2 | 92.8 | 47.7 | 79.2 | 75.5 | 82.5 |
| no erasure training (clean clips only) | 3.7 | – | 5.2 | – | 26.3 | – |
| MemJEPA-B | 93.0 | 94.7 | 52.0 | 78.8 | 79.0 | 83.8 |
| state frozen inside candidates | 57.0 | 76.0 | 19.2 | 24.5 | 75.2 | 82.7 |
| MemJEPA-S | 93.2 | 93.0 | 57.0 | 72.0 | 83.3 | 87.0 |

B on TwoRoom ($+4.3\,[+0.7,\ +8.2]$) and is level on PushT and Reacher, while with a full view MemJEPA-B is ahead by 1.7 on Reacher and 6.8 on PushT and behind by 3.2 on TwoRoom (each interval excludes zero).

## 5 Conclusion

We introduced MemJEPA, which carries a recurrent belief through the imagination of a frozen-encoder JEPA planner and trains it on clips in which the controlled entity is erased from the input but kept in the prediction targets. We showed that a fixed-window planner cannot react to the motion of an entity it no longer sees when that motion leaves no trace in its window, and that erasure training against clean targets provides the learning signal a useful carried state needs. Our results suggest that memory for latent planning is as much a question of the training signal as of the architecture, since neither erasure without a carried state nor a carried state without erasure leaves the memoryless floor, while their combination restores planning. Empirically, on reaching, pushing and navigation with LeWM's frozen encoder, MemJEPA raises success with the actor hidden from 13% to about 75% against a memoryless twin trained with the same erasure, while matching or exceeding it with the actor in view, and unlike a last-visible cache or a fixed context window it keeps working once the entity moves or has been hidden longer than the window holds. The hidden-view gains carry over, at a smaller size, to a DINO-WM backbone. As the belief is a point estimate that the planner never doubts, one exciting direction for future work is a belief that represents its own uncertainty and lets the planner act to reduce it. On a practical note, we believe that evaluating erasure-trained beliefs on real robots under self-occlusion, and with encoders trained jointly with the memory, can shed further light on how far latent planners can act on what they no longer see.

### Reproducibility statement

The code, per-episode records, case files, entity masks and held-out manifests are released with the supplementary material, with checksums. Every table cell, except those marked, is recomputed from the per-episode records by a re-tally script written separately from the experiment code. Appendix I gives the full protocol, hyperparameters and compute.

## REFERENCES


Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025. arXiv:2506.09985.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics, 2025. arXiv:2511.08544.

Samuel Barbeau, Simon Roy, Giovanni Beltrame, Christian Desrosiers, and Nicolas Thome. Latent goal prediction from language for model-based planning, 2026. arXiv:2606.20627.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, 2015. arXiv:1506.03099.

Niccolò Caselli, Francesco Massafra, Samuele Punzo, Salvatore Lo Sardo, Ippokratis Pantelidis, and Sathya Kamesh Bhethanabhotla. Mind the gap: Promises and pitfalls of hierarchical planning in LeWorldModel, 2026. arXiv:2607.12547.

Letian Cheng, Qi Zhang, and Yisen Wang. SAGE: Subgoal-conditioned action generation for latent world model planning, 2026. arXiv:2607.17973.

Egor Cherepanov, Nikita Kachaev, Daniil Zelezetsky, Aydar Bulatov, Artem Pshenitsyn, Yuri Kuratov, Alexey Skrynnik, Aleksandr I. Panov, and Alexey K. Kovalev. $\mu$VLA: On recurrent memory for partially observable manipulation in VLA models, 2026. arXiv:2606.12497.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023. arXiv:2303.04137.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014. arXiv:1406.1078.

Fei Deng, Junyeong Park, and Sungjin Ahn. Facing off world model backbones: RNNs, transformers, and S4. In *NeurIPS*, 2023. arXiv:2307.02064.

Yurui Du, Pinhao Song, Yutong Hu, and Renaud Detry. ELVIS: Ensemble-calibrated latent imagination for long-horizon visual MPC, 2026. arXiv:2605.04709.

C. Daniel Freeman, Luke Metz, and David Ha. Learning to predict without looking ahead: World models without forward prediction. In *NeurIPS*, 2019. arXiv:1910.13038.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023. arXiv:2312.00752.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *NeurIPS*, 2020. arXiv:2008.07669.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022a. arXiv:2111.00396.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *NeurIPS*, 2022b. arXiv:2206.11893.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019. arXiv:1811.04551.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. In *NeurIPS*, 2022. arXiv:2206.04114.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2023. arXiv:2301.04104.

Yixuan Huang, Jialin Yuan, Chanho Kim, Pupul Pradhan, Bryan Chen, Fuxin Li, and Tucker Hermans. Out of sight, still in mind: Reasoning and planning about unobserved objects with video tracking enabled memory models. In *ICRA*, 2024. arXiv:2309.15278.

I-Tak Ieong, Ruizhi Feng, Zhaoyang Lu, Yifei Cao, Jiayao Zhao, Leon Li, Senhua Zhu, and Wenbo Ding. GLAM: Training a latent world model over global spatiotemporal memory for active exploration and navigation, 2026. arXiv:2609.14561.

Markus Karmann, Shile Li, Christian Internò, Bruno Andreis, David Klindt, Randall Balestriero, Jindong Gu, Philip Torr, Qi Zhang, Peng-Tao Jiang, Hao Zhang, Bo Li, and Onay Urfalioglu. MotionJEPA: Preventing temporal feature collapse by capturing visual changes in latent space, 2026. arXiv:2609.23881.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels, 2026. arXiv:2603.19312.

Sergi Masip, Jonathan Swinnen, Yutong Hu, Renaud Detry, and Tinne Tuytelaars. FF-JEPA: Long-horizon planning in world models with latent planners, 2026. arXiv:2606.09311.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-JEPA: Learning world models through object-level latent masking. In *ICML*, 2026. arXiv:2602.11389.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *ICML*, 2023. arXiv:2303.06349.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *ICLR*, 2025. arXiv:2410.20092.

Jurgis Pasukonis, Timothy Lillicrap, and Danijar Hafner. Evaluating long-term memory in 3D mazes. In *ICLR*, 2023. arXiv:2210.13383.

Phu Pham and Aniket Bera. Latent energy action planning with world models, 2026. arXiv:2609.03294.

Marco Pleines, Matthias Pallasch, Frank Zimmer, and Mike Preuss. Memory Gym: Towards endless tasks to benchmark memory capabilities of agents. *Journal of Machine Learning Research*, 26(6): 1–40, 2025. arXiv:2309.17207.

Santosh Kumar Radha and Oktay Goktas. UWM-JEPA: Predictive world models that imagine in belief space, 2026. arXiv:2605.25313.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer, 2004.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models. In *ICLR*, 2024. arXiv:2403.04253.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *CoRL*, 2022. arXiv:2206.14244.

Rishi Shah and Rishav Shrestha. Rollout-decoded reconstruction for long-horizon prediction in latent world models, 2026. arXiv:2608.25017.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *NeurIPS*, 2025. arXiv:2502.14819.

Haoyun Tang, Haodong Cui, Keyao Xu, Kun Wang, and Zhandong Mei. HaM-World: Soft-Hamiltonian world models with selective memory for planning, 2026. arXiv:2605.05951.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite, 2018. arXiv:1801.00690.

Manuel Traub, Frederic Becker, Sebastian Otte, and Martin V. Butz. Learning object permanence from videos via latent imaginations. In *ICANN*, 2024. arXiv:2310.10372.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model, 2026. arXiv:2606.32026.

Yilun Wu, Yunjian Zhang, Aobo Li, Mujiangshan Wang, Haitao Wu, and Aqiang Zhang. FIRM-WM: State-factorized factual–interventional recurrent modeling for reward-free visual planning, 2026. arXiv:2609.22816.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models, 2026. arXiv:2604.03208.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-JEPA: Subspace Gaussian regularization for stable end-to-end world models, 2026. arXiv:2605.09241.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025. arXiv:2411.04983.

Todd Y. Zhou and Daniel Zhang. Learning counterfactual world models for embodied reasoning under partial observability, 2026. arXiv:2609.05834.

## A LONG HORIZONS: THE WAYPOINT READOUT

This appendix tests the waypoint readout, which fires only while more than 25 steps remain and so never acts in the main results. Table 5 and Figure 6 evaluate it at goal 75 (two replans) against four bars: memoryless, the last-visible cache, memoryless without the head in full view (non-inferiority at $-5$, in a column that includes the head's own gain), and the same model without the head. Because the head reads the frame, it helps only while the goal's entity is visible. We chose the switched head and its two-of-three criterion post hoc, on the one-step own-prediction model.

On PushT the goal's entity is the T-block, which stays visible while the pusher is hidden. The first three bars pass on this model; the fourth was measured on the one-step own-prediction variant, where the head adds 12.0 points (Table 5). Routing by the pusher instead is an ablation (supplementary material, Section S2). On Reacher the goal depends on the arm, so the head is on only while the arm is visible. Its full-view gain also lifts the memoryless model, so the gain belongs to the head; left on with the arm hidden, it costs 34.8 points. On TwoRoom it adds nothing.

The switch is a linear probe on the frozen frame latent, trained on erasure labels from training episodes. On Reacher it matches the oracle on every head-eligible decision, so the gain is unchanged (Table 11); PushT and TwoRoom use the oracle. Past the training clip, at a horizon of 10 blocks, the belief still beats memoryless but loses most of its horizon-5 success (supplementary material, Section S21).

## B FULL MODEL AGAINST THE BASE

This appendix checks that the components add up and that the belief costs nothing with a full view (Table 6). On fully observed PushT (one fresh draw, goals 50 and 75) the belief-site models trail a memoryless model with a frame head by 20–23 points. The erased-clip recipe and the readout site cause this gap, not the belief: with a frame head the belief model matches released LeWM, and on LeWM's own recipe the belief costs nothing. A memoryless model on that recipe is level with released LeWM (supplementary material, Sections S2 and S21).

On Reacher the evaluation goal lies within 25 steps, so the waypoint never fires and the readout is tested only in the far-goal, full-view column, where it carries the gain: removing the readout or

Table 5: Goal 75: belief model with the waypoint head routed by the visibility of the goal-relevant entity (held-out sets, entity hidden unless stated; points, 95% CI). *One-step own-prediction variant. †Entity visible. §Training-split episodes (all goal-75 TwoRoom episodes are in the world models' training split; the head was fitted without them).

| | − memoryless | − cache | full view | head − no head |
|---|---|---|---|---|
| PushT, pusher hidden | +21.8 [+16.5, +27.3] | +15.2 [+9.3, +21.2] | +26.0 [+19.5, +32.5] | +12.0* [+6.0, +18.0] |
| Reacher, arm hidden | +83.0 [+79.8, +86.0] | +70.7 [+66.8, +74.5] | +8.0 [+5.7, +10.5] | +8.0† [+5.8, +10.5] |
| TwoRoom, agent hidden§ | +25.8 [+20.2, +31.5] | +15.7 [+10.8, +20.8] | +4.2 [−1.7, +10.0] | +2.3† [−3.5, +8.3] |

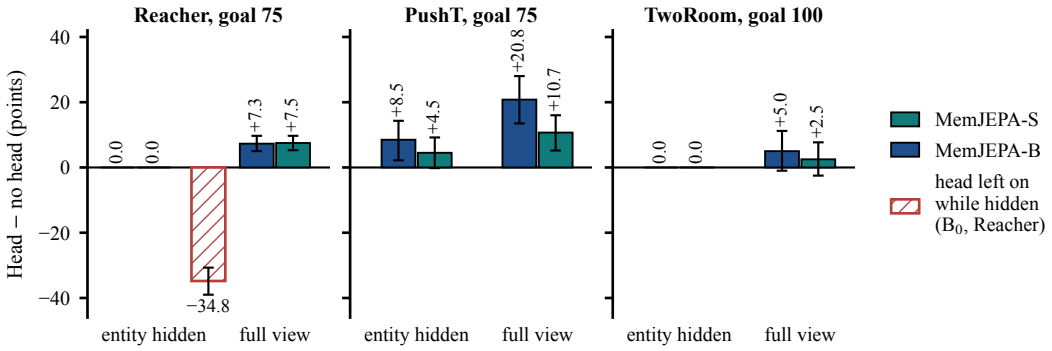



Figure 6: Waypoint head: head − no head for MemJEPA-B and MemJEPA-S, goal's entity hidden and visible (goal 75, TwoRoom 100; held-out; 95% intervals as in Table 8). Hatched: $B_0$ with the head left on while the arm is hidden.

freezing the read costs most, and removing the belief nothing. With the arm hidden, $B_0$ lifts the memoryless base from 7.5 to 91.2% on held-out set S2 (Table 16; Table 6 uses per-seed validation sets), a lift that holds over five seeds. Input reading and own-prediction training at the output site are two routes to the same gain, and combined they do worse (Section 4.2).

Table 6: Full model against the base, and leave-one-out (success %). P: published (Maes et al., 2026; Pham & Bera, 2026). R: released checkpoint, our evaluator. T: same data, recipe and initialization, no memory. $B_0$: the model of Section 3 (input read, advanced); its row here adds the switched waypoint. Far: Reacher, goal 100; g20/g60: occluded-PushT recall construct (block hidden), goals 20/60 (full view: goal 50). Bold: best hidden. ‡Identical to an existing arm. ♭LeWM's own recipe, goal 50. ♮Official settings. –: not run.

| | full view | | | | | entity hidden | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | PushT | Reacher | TwoR. | Cube | far | g20 | g60 | Reacher | TwoR. |
| (P) LeWM+CEM, published | 96 / 88 | 86 / 75 | 87 / 85 | 74 / 62 | – | – | – | – | – |
| (R) LeWM+CEM, ours | 42.0 | 95.8 | 81.5 | 59.0♮ | 85.0 | – | – | 8.5 | 19.0 |
| + frame-site readout | 57.5 | – | – | – | 99.0 | – | – | – | – |
| (T) fine-tune, no memory | 40.5♭ | 95.7 | 84.7 | – | 83.7 | 21.3 | 26.8 | 10.2 | 28.5 |
| $B_0$ + switched waypoint | 42.7♭ | 95.7 | 82.5 | – | – | 80.3 | 41.3 | **94.3** | **75.5** |
| + own-prediction training | – | 96.0 | 81.2 | – | – | – | – | 90.3 | 74.8 |
| Output site + own-pred. (below) | – | 94.5 | – | – | 98.0 | – | – | 93.5 | – |
| − belief (twin + frame head) | – | 95.7‡ | – | – | 99.7 | – | – | 10.2‡ | – |
| − own-prediction training | – | 94.8 | – | – | 97.3 | – | – | 76.2 | – |
| − advancing | – | 28.3 | – | – | 46.7 | – | – | 26.8 | – |
| − waypoint readout | – | 94.5‡ | – | – | 84.3 | – | – | 93.5‡ | – |
| frame-site readout | – | – | – | – | 95.3 | – | – | – | – |

## C MEMJEPA-S COMPARISON: SUCCESS RATES

This appendix gives the success rates behind Section 4.3 (Table 7). MemJEPA-S had one configuration per seed and one retry, fixed in advance, for a failed parity check; MemJEPA-B's read site and schedule were tuned on development sets. MemJEPA-S's state transfers to an unseen erasure onset time and the belief's does not; neither transfers to all-black frames. A deterministic recurrent predictor trained from scratch recovers most of MemJEPA-S's Reacher gain (Table 8), so the recurrent state carries most of it; the transformer with the additive belief sits a few points above that predictor. Supplementary Section S21 gives the rates.

Table 7: Held-out tracking sets (success %), all arms re-run on one machine per environment; no arm sees visibility. All arms share data, clips, training budget, frozen encoder and planner (Table 15). Bold: the case-bootstrap CI of the difference to the next-best arm excludes zero; this interval omits seed variance.

| | Reacher (arm) | | PushT (pusher) | | TwoRoom (agent) | |
|---|---|---|---|---|---|---|
| Arm | hidden | full | hidden | full | hidden | full |
| Memoryless twin | 7.5 | 93.8 | 3.3 | 51.7 | 29.3 | 84.3 |
| RSSM (Hafner et al., 2019), standard training | 8.5 | 94.2 | 7.8 | 74.3 | 27.8 | 81.0 |
| MemJEPA-S | 93.2 | 93.0 | **57.0** | 72.0 | **83.3** | 87.0 |
| $B_0$ | 91.0 | 92.7 | 47.7 | 79.2 | 74.5 | 82.3 |

## D BELIEF FAMILY: CARRIER AND READ COMBINATIONS

This appendix compares the carriers, read sites and training additions we tried before settling on the two final members: the plain recipe-trained stateful predictor (MemJEPA-S) and the belief with imagination-consistent training (MemJEPA-B).

We also tried a diagonal linear recurrence in the S4D and LRU form (Gu et al., 2022b; Orvieto et al., 2023) (input-independent decay, unlike Mamba's (Gu & Dao, 2023)), $m_t = \lambda \odot m_{t-1} + \sqrt{1-\lambda^2} \odot Bx_t$, with $\lambda = \exp(-\exp(\nu))$ initialized in $[0.90, 0.999]$, in place of the GRU cell; it is width-matched, not parameter-matched. With the arm hidden it trails the GRU (Table 8); with a full view the two cells are interchangeable after a 20-step warm-up (supplementary material, Section S15). We report the GRU in the main text. Each choice rests on a contrast against a same-build re-run of the GRU belief, given in the supplementary material (Section S21) rather than in Table 8. An LSTM ties the GRU on Reacher and TwoRoom but trails it on PushT, so we keep the GRU. A diagonal linear carrier keeps about 60% of the Reacher gain. The mix gate, a learned convex mix of frame latent and memory, trails the additive read only on PushT, and MemJEPA-S's stochastic cell, used as the belief carrier, is below the GRU belief everywhere except in full view on Reacher and TwoRoom (Tables 8 and 9).

Imagination-consistent training adds a few hidden points on PushT and TwoRoom, is level on Reacher, costs no full-view success and alone gives no all-black coverage. The waypoint head passes on Reacher and on PushT, where the T-block is never erased, so the head also acts with the pusher hidden; there MemJEPA-B with the head leads MemJEPA-S. The head fails on TwoRoom for both designs and stays out of both stacks as a task-dependent full-view addition. On the MemJEPA-B stack the mixed erasure schedule trades PushT hidden success for all-black coverage (Table 17 gives the module alone) and ties imagination-only training on TwoRoom; for MemJEPA-S, imagination-consistent training helps only on TwoRoom and costs PushT. Neither the schedule nor this term enters MemJEPA-S's recipe, and the schedule stays out of MemJEPA-B's.

## E PLDM: DETAILS

This appendix gives the details of the PLDM port (Sobal et al., 2025). Our PLDM reaches 85.7% on PLDM's two-rooms protocol against 97.8% published; the released code lacks the $K = 5$ predictor ensemble, and our diagnostics found no data or evaluation cause. The contrasts below are therefore

Table 8: The belief family: success difference against memoryless LeWM (points, 95% CI; held-out, 200 cases × 3 seeds). $B_0$: MemJEPA-B without the imagination-consistent term. †Re-run on one software build. ‡/§PushT after/before a training-code fix. *Full-view bound below −5 on PushT. ◇No belief. ♮Absolute success. ♯Head minus no head, goal 75 (TwoRoom 100).

| *Entity hidden* Member | Reacher | PushT | TwoRoom |
|---|---|---|---|
| Twin, v2 recipe◇ | +2.8 [+0.3, +5.3] | +0.2 | +23.5 [+18.0, +28.8] |
| $B_0$ | +83.7 [+79.3, +87.7] | +44.3 [+38.3, +50.5] | +47.0 [+40.3, +53.5] |
| $B_0$, GRU, mix gate† | +81.5 [+78.2, +84.7] | +41.8 [+37.7, +46.0] | +45.5 [+41.0, +50.0] |
| MemJEPA-B‡ | +85.5 [+82.7, +88.2] | +48.7 [+44.5, +52.8] | +49.7 [+45.2, +54.2] |
| input-mix read§ | – | – | +48.8 [+44.5, +53.2] |
| + mixed erasure schedule | +84.3 [+81.3, +87.2] | +21.2 [+16.3, +26.2] | +48.3 [+41.5, +54.8] |
| + waypoint head♯ | +0.0 | +8.5 [+2.2, +14.3] | +0.0 |
| Deterministic stateful pred.* | +78.0 [+74.7, +81.2] | +27.0 [+23.0, +31.0] | +56.0 [+51.8, +60.2] |
| $B_0$, LSTM (0.89M)† | +81.7 [+78.5, +84.7] | +36.7 [+32.7, +40.5] | +44.7 [+40.0, +49.3] |
| $B_0$, stochastic cell | +6.3 [+3.2, +9.5] | +26.8 [+23.2, +30.5] | +38.5 [+33.7, +43.2] |
| $B_0$, diagonal linear (313k) | +50.5 [+46.0, +54.8] | – | – |
| MemJEPA-S | +85.7 [+81.7, +89.5] | +53.7 [+48.3, +58.8] | +54.0 [+47.7, +60.2] |
| + waypoint head♯ | +0.0 | +4.5 [−0.2, +9.2] | +0.0 |
| + imagination-consistent training | 92.5%♮ | 48.7%♮ | +55.8 [+51.5, +60.2] |
| + mixed erasure schedule♮ | 92.3% | 34.3% | 85.5% |

| *Full view* Member | Reacher | PushT | TwoRoom |
|---|---|---|---|
| Twin, v2 recipe◇ | 94.5%♮ | +2.0 | −0.5 [−2.5, +1.5] |
| $B_0$ | −1.0 [−2.8, +0.8] | +27.5 [+22.0, +33.0] | −2.2 [−4.3, −0.2] |
| $B_0$, GRU, mix gate† | −0.3 [−2.0, +1.3] | +22.0 [+17.8, +26.2] | −2.5 [−4.7, −0.3] |
| MemJEPA-B‡ | +0.8 [−0.8, +2.5] | +27.2 [+23.2, +31.3] | −0.5 [−2.3, +1.3] |
| input-mix read§ | – | – | −0.3 [−2.2, +1.5] |
| + mixed erasure schedule | −0.2 [−1.8, +1.5] | +17.2 [+11.3, +23.2] | +0.5 [−1.2, +2.2] |
| + waypoint head♯ | +7.3 [+5.0, +9.7] | +20.8 [+13.5, +28.0] | +5.0 [−1.0, +11.2] |
| Deterministic stateful pred.* | −1.7 [−3.5, +0.2] | −1.3 [−6.0, +3.3] | +4.3 [+1.7, +7.0] |
| $B_0$, LSTM (0.89M)† | −0.3 [−2.0, +1.3] | +20.5 [+16.3, +24.8] | −2.5 [−4.7, −0.3] |
| $B_0$, stochastic cell | +0.2 | +9.8 [+5.5, +14.2] | −1.8 [−4.0, +0.2] |
| $B_0$, diagonal linear (313k) | +0.2 [−1.7, +2.0] | – | – |
| MemJEPA-S | −0.8 [−2.7, +0.8] | +20.3 [+14.3, +26.3] | +2.7 [−0.3, +5.7] |
| + waypoint head♯ | +7.5 [+5.3, +9.7] | +10.7 [+5.2, +16.0] | +2.5 [−2.5, +7.7] |
| + imagination-consistent training | 92.8%♮ | 65.0%♮ | +2.0 [−0.2, +4.2] |
| + mixed erasure schedule♮ | 92.8% | 57.3% | 85.5% |

Table 9: Carrier and read combinations on Reacher: belief − memoryless (points, held-out, 200 cases × 3 training seeds, 95% pair-clustered bootstrap intervals).

| Reacher | hidden | full |
|---|---|---|
| LSTM carrier, additive read | +81.7 [+78.5, +84.7] | −0.3 [−2.0, +1.3] |
| LSTM carrier, mix gate | +83.2 [+80.0, +86.2] | +0.7 [−1.2, +2.5] |
| Diagonal linear, additive read | +50.5 [+46.0, +54.8] | +0.2 [−1.7, +2.0] |
| Diagonal linear, mix gate | +49.3 [+45.0, +53.7] | −3.5 [−5.7, −1.3] |

paired within a reproduction that does not reach parity. All use the PLDM held-out sets (200 cases × 3 training seeds); Section S21 of the supplementary material gives them with intervals.

With the entity hidden, a belief under C1 and C2 beats a recipe-matched memoryless twin in all three environments. On TwoRoom it also beats a stale-agent cache, which we ran there only, and matches its twin with a full view; zeroing or swapping the state raises its masked-window loss more than twentyfold, so the read is live. With a full view the belief also gains on PushT and Reacher,

although the PushT base is weak (34% on the PushT parity check against 78% reported for PLDM by Maes et al. (2026)).

MemJEPA-S ties the PLDM belief in all three environments and on PushT beats the twin in both views. It has 1,064,170 parameters against 893,725 for the PLDM belief, not re-matched, and its CUDA-graph trainer matches the eager trainer's final weights to $6 \times 10^{-7}$.

## F DINO-WM: DETAILS

This appendix tests whether memory transfers to a second base world model. DINO-WM plans with frozen DINOv2 patch tokens and a per-patch loss; its encoder is not JEPA-trained, so it is a joint-embedding predictive planner (Section 1) but not a LeWM-like one. Each memory arm is compared with a trained memoryless twin with the belief's architecture, size, read site and loss and its state zeroed at every step; there is no cache arm. MemJEPA-B freezes the predictor and adds the imagination-consistent term and, when the entity covers under 5% of patches, an entity-weighted loss ($\lambda = 10$); MemJEPA-S replaces the predictor. Sections S19 and S21 of the supplementary material give parity checks, stock-predictor intervals, planner seeds and the shared design. Memory

Table 10: DINO-WM as the base world model (held-out sets, 200 cases $\times$ 3 training seeds): belief − trained memoryless twin, success points with 95% CIs, and absolute success of the stock predictor, twin, MemJEPA-B and MemJEPA-S. $^\sharp$Arm hidden for 20 strided frames. $^\dagger$One planner seed for all training seeds.

| Environment | View | MemJEPA-B − twin | MemJEPA-S − twin | stock / twin / B / S (%) |
|---|---|---|---|---|
| Reacher | hidden | $+13.0\,[+9.0, +17.0]$ | $+40.0\,[+35.0, +44.8]$ | 5.8 / 6.8 / 19.8 / 46.8 |
| | full | $-5.2\,[-8.3, -2.0]$ | $-26.5\,[-31.7, -21.3]$ | 70.8 / 82.0 / 76.8 / 55.5 |
| | $L = 20$, hidden$^\sharp$ | $+4.2\,[+0.5, +7.8]$ | $+17.2\,[+13.0, +21.3]$ | 5.3 / 6.8 / 11.0 / 24.0 |
| PushT | hidden | $+5.0\,[+1.8, +8.3]$ | $+19.3\,[+14.8, +23.8]$ | 1.0 / 3.0 / 8.0 / 22.3 |
| | full | $+7.7\,[+2.5, +12.8]$ | $-32.3\,[-38.0, -26.7]$ | 88.5 / 55.7 / 63.3 / 23.3 |
| TwoRoom | hidden | $+34.2\,[+28.0, +40.2]$ | $+43.2\,[+38.0, +48.3]^\dagger$ | 21.2 / 40.2 / 74.3 / 83.3 |
| | full | $+0.0\,[-0.5, +0.5]$ | $-15.2\,[-19.8, -10.8]^\dagger$ | 99.5 / 99.8 / 99.8 / 84.7 |

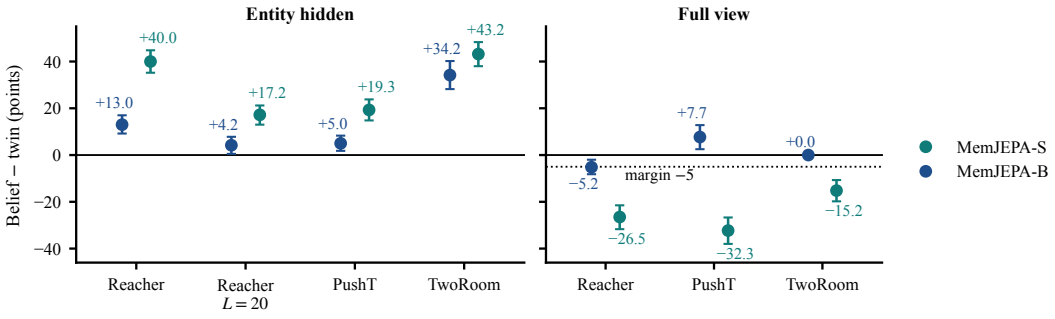



Figure 7: DINO-WM as the base world model: belief − trained memoryless twin (points; held-out, 200 cases $\times$ 3 training seeds; 95% case-clustered bootstrap intervals), entity hidden (left) and full view (right). Dotted line: the $-5$ point full-view non-inferiority margin.

helps on every task once the entity is hidden (Table 10, Figure 7). On TwoRoom, whose base we trained with the authors' code (parity 99 of 100), MemJEPA-B recovers most of the stock predictor's hidden-view drop at no full-view cost; on Reacher it fails the 5-point full-view margin, and on PushT it gains in both views. At $L = 20$ both memory arms keep smaller leads over the twin and stay far below their LeWM counterparts (Table 14). The backbones differ in where memory enters, and the gap follows: on LeWM the small predictor is fine-tuned with the belief, while on DINO-WM MemJEPA-B must steer a large frozen ViT through its input alone. MemJEPA-S, which replaces the predictor, has the largest hidden gains on DINO-WM but pays a full-view cost there that it does not pay on LeWM (Table 7). Recurrence alone does not help (Table 7, standard training); erasure training does, on both backbones. We did not test fine-tuning DINO-WM's predictor with the belief.

The entity weighting comes from PushT, where the pusher covers 3.1% of patches and its error is lost in the mean patch loss. Unweighted, the state encodes the pusher, yet zeroing or swapping it leaves the loss unchanged and the belief ties its twin. The belief helps only where the objective reads it, so only TwoRoom, where the agent covers more patches, trains unweighted. Any read trained into the frozen PushT predictor also costs full-view success; a control never trained on erased clips shows that the trained read, not memory, causes most of this cost, and that erasure training produces the hidden gain (supplementary material, Section S19).

## G Deviations, negative ports and manipulation

This appendix lists where our setup departs from the published ones (Table 11) and which ports failed. Two ports stopped before we evaluated any memory arm. On Memory Gym's Searing Spotlights a ridge probe on the frozen latent locates the agent only to 23.9 px, against the 6 px we required. On DINO-WM's Wall our trained base reached 23.3% (35 of 150 episodes) against the released planner's 96%; the released checkpoint passed parity, but its read did not change the prediction loss. Section S18 of the supplementary material covers the robomimic port. Both failures come from planning in a frozen pooled latent. A small entity, such as a Memory Gym agent or a fingertip, does not survive the encoder unless it covers enough patches or the loss is entity-weighted (the 5% rule of Section 4). Latent goal distance also expresses "be here" tasks, not contact events such as a grasp.

Table 11: Deviations from the published setups, with reason and measured effect.

| Deviation | Reason | Measured effect |
|---|---|---|
| PLDM reproduction | the authors' predictor ensemble is not in the released code | 87, 83, 87% against 97.8% published on their task; belief − twin contrasts are within our reproduction |
| Cube | released checkpoint below its published figure on our draw | 59.0% [52.1, 65.6] against 74%; Cube not used |
| Reacher context and planner | our models plan from three frames, the official evaluator from one; CEM 10 iterations on Reacher, 30 elsewhere | released LeWM 80.7% [73.6, 86.2] with one frame, 93.3% [88.2, 96.3] with three |
| Waypoint visibility switch | a learned linear probe on the frame latent (verified on Reacher only) | equals the oracle flag on 19,450 of 19,450 decisions; 0 of 600 episodes differ |
| Memory Gym Searing Spotlights | attempted as an established partially observed benchmark | the pooled latent does not encode the agent, so no held-out read was run |

## H Read rule: argument and further Reacher controls

This appendix rules out other explanations of the gain and gives the argument for reading and advancing the belief as we do.

### Baselines and mechanism, in full

Table 12 pairs each alternative explanation of the gain with its control. The 10- and 40-frame windows are the memoryless-history baseline, and the severed-recurrence module (state zeroed every step; on DINO-WM, the twin) is the parameter-matched control. A 10- to 15-frame window ties the belief on TwoRoom, where MemJEPA-S leads both, but trails it on Reacher and PushT except past its training clip (supplementary material, Section S20). On Reacher the post-hoc 10-frame window also beats the 40-frame window, which we cannot explain. At 20 hidden Reacher frames MemJEPA-B leads the 40-frame window by $+6.3\,[-1.0,\ +13.7]$ (Table 14; supplementary material, Section S20); at 33 frames the window leads the belief, 45.5 against 24.8% (Section S12).

The belief holds the hidden entity, and the planner uses it. One to five steps after hiding, a probe decodes the Reacher fingertip from the belief of the own-prediction models at 0.029 m error, against 0.114 m from the last visible latent. The state forms only where the objective is sensitive to the entity: zeroing or swapping it raises the masked-window loss sharply on LeWM and PLDM (supplementary

material, Section S12), and swapping or freezing it (Table 12) shows that the planner uses its content. MemJEPA-S can win because it receives the same erased-clip signal, so the training conditions carry the gain.

Table 12: Alternative explanations of the gain, each with its control (Reacher, arm hidden, held-out unless stated; 95% intervals). *MemJEPA-B and -S on all three held-out sets, post hoc (swap: hidden only; full view in parentheses).

| Control | Hypothesis it rules out | Hidden-view result |
|---|---|---|
| *Memory is needed* | | |
| More candidates (Table 19) | planning compute substitutes | memoryless 9.5% at 1200; $B_0$ 86.3% at 150 |
| Full view, nothing hidden | the belief helps regardless of occlusion | $-1.0\,[-2.8,\ +0.8]$ against memoryless (full view) |
| *The gain is the belief's content, read at the input and advanced* | | |
| Last-visible cache | a memory of the last frame suffices | 25.3% against 91.2% for $B_0$ |
| Severed recurrence (parameter-matched control) | extra parameters, not memory | $-1.5\,[-3.8,\ +0.7]$ against memoryless |
| Same belief, predictor frozen | the fine-tune does the work | $+83.0\,[+79.8,\ +86.0]$ against memoryless (90.5%; full $+0.2\,[-1.7,\ +2.0]$); vs fine-tuned: PushT $+0.2\,[-5.0,\ +5.3]$ (47.8 vs 3.3% memoryless), TwoRoom $-2.0\,[-6.8,\ +2.7]$ (71.7 against 73.7%: $B_0$ re-run on the same software build) |
| Carried state swapped between paired episodes* | the belief's content is unused | B 7.7, S 7.2% ($-85.3\,[-89.7,\ -80.7]$, $-86.0\,[-90.3,\ -81.2]$); PushT 4.3, 2.8%; TwoRoom 22.2, 21.2% (memoryless 7.5, 3.3, 29.3%) |
| State frozen inside candidates* | advancing in imagination is incidental | $-36.0\,[-41.8,\ -30.3]$ (full $-18.7\,[-23.3,\ -14.0]$); PushT $-32.8\,[-39.2,\ -26.3]$ (full $-54.3\,[-60.2,\ -48.5]$); TwoRoom $-3.8\,[-7.8,\ +0.2]$ (full $-1.2\,[-3.7,\ +1.3]$), against MemJEPA-B |
| RSSM, standard training (Section 4.3) | recurrence alone suffices | $+1.0\,[-2.8,\ +4.8]$ against memoryless |
| C1 removed: same module, clean clips only | erasure training is incidental | 3.7%, $-87.3\,[-89.8,\ -84.7]$ against the erasure-trained module; PushT 5.2%, $-44.7\,[-50.7,\ -38.8]$; TwoRoom 26.3%, $-47.0\,[-54.2,\ -39.7]$ (PushT and TwoRoom against same-build re-runs) |
| B's input read removed: output-site read | the read site does not matter | $-15.7\,[-19.2,\ -12.2]$ against the input site |
| C2 removed, output-site belief (earlier-used set) | advancing need not follow imagination | every five steps $-23.7\,[-28.5,\ -18.8]$; action only $-27.5\,[-32.5,\ -22.5]$ |

We fixed the removal controls before running any of them, though after the held-out sets had been evaluated. Against the memoryless twin the frozen read keeps part of the gain on Reacher and PushT (supplementary material, Section S21, with per-seed counts).

### WHY A WINDOW PREDICTOR CANNOT PLAN THROUGH OCCLUSION

Let $o_t$ be the rendered frame, $z_t = e(o_t)$ its latent (the $\tilde{z}_t$ of Eq. (2)), and $s_t$ the state of the task entity. A hidden entity may still leave traces in $o_t$ through coupling; the argument needs only equal windows. A $k$-window predictor $\hat{z}_{t+1} = P\big(f(z_{t-k+1:t}, a_{t-k+1:t})\big)$ imagines a rollout from the last $k$ latents and actions, and a sampling planner such as CEM scores each candidate $a_{t:t+H-1}$ by $J(a) = \phi(\hat{z}_{t+1:t+H}, z_g)$, with $z_g$ fixed per episode, and returns a distribution that depends on the candidates only through their scores and on randomness independent of the history.

**Proposition 1** (Window blindness)**.** *If two histories share the goal and the last $k$ observations and actions, with the entity hidden throughout, the $k$-window planner returns the same action distribution on both. Hence, on episodes in which the entity has been hidden for at least $k$ frames, its success rate is at most that of the best policy whose action at each replanning time is measurable with respect to the goal and the last $k$ frames and actions.*

*Proof.* Each imagined step reads only the window, earlier imagined latents and candidate actions, so equal windows and goals give every candidate the same rollout and score, and the planner the same distribution. It is thus a policy measurable with respect to the goal and the window, bounded by the best such policy. □

A planner that keeps the latent of the last frame showing the entity is bounded likewise by that frame and the actions since, and its estimate of $s_t$ goes stale when the entity moves in ways those actions do not determine.

For training, input frames are erased after a visible prefix of $k_{\text{vis}}$ frames and the targets are the clean next latents $z_{t+1} = e(o_{t+1})$. Write $H_t = (\tilde{o}_{1:t}, a_{1:t})$ for the erased history and $W_t = (\tilde{o}_{t-k+1:t}, a_{t-k+1:t})$ for its last $k$ frames and actions; a model with a state reads a function of $H_t$, a $k$-window model a function of $W_t$, and both minimize $\mathbb{E}\,\|z_{t+1} - f\|^2$.

**Proposition 2** (What erasure training makes the state carry)**.** *The squared error is minimized by $m_H = \mathbb{E}[z_{t+1} \mid H_t]$ over functions of $H_t$ and by $m_W = \mathbb{E}[z_{t+1} \mid W_t]$ over functions of $W_t$, and*

$$\mathbb{E}\,\|z_{t+1} - m_W\|^2 - \mathbb{E}\,\|z_{t+1} - m_H\|^2 \;=\; \mathbb{E}\,\|m_H - m_W\|^2 \;=\; \mathbb{E}\big[\operatorname{tr}\operatorname{Var}(m_H \mid W_t)\big] \;\geq\; 0.$$

*The gap is zero (i) when the entity is visible in every input frame, so that $s_t$ is a function of $\tilde{o}_t$, and $z_{t+1}$ depends on the past only through $(s_t, \tilde{o}_t, a_t)$, and (ii) when the targets are erased too and the rest of the scene evolves independently of the entity, so that the erased target depends on the past only through $(\tilde{o}_t, a_t)$.*

*Proof.* For any function $f$ of $W_t$, and hence of $H_t$, the residual $z_{t+1} - m_H$ is orthogonal to $m_H - f$, so $\mathbb{E}\,\|z_{t+1} - f\|^2 = \mathbb{E}\,\|z_{t+1} - m_H\|^2 + \mathbb{E}\,\|m_H - f\|^2$. By the tower property the second term is minimized by $f = \mathbb{E}[m_H \mid W_t] = m_W$, which gives the identity. In case (i) $s_t$ is a function of $\tilde{o}_t$, and in case (ii) the target does not involve $s_t$, so the conditional mean of the target is a function of $(\tilde{o}_t, a_t)$, hence of $W_t$, and the variance is zero. □

Under the dependence assumption of (i) but with the entity erased throughout the window, its state fixed by the visible prefix and the actions since, the gap is the expected variance of $m_H$ under the posterior of $s_t$ given the window. It is zero when the hidden entity does not move the mean of the clean next latent, and positive exactly when, with positive probability, the window leaves $s_t$ uncertain among states that give different means, since a conditional variance is positive exactly when the conditional distribution is not degenerate. With coupling, as on PushT where the hidden pusher moves the block, an erased target keeps the part of the gap that the coupling carries.

The propositions establish necessity, not sufficiency: information from before the window is needed whenever the best $k$-window policy falls short, a recurrent belief (Eq. (1)) depends on the whole history and so escapes Proposition 1, and erasure training against clean targets gives it something to learn. Whether a trained state recovers enough of $s_t$, and whether the planner reads it, are the empirical questions of Section 4.

Figure 5 collects the mechanism controls; all ablations in one table are in the supplementary material (Section S21). For the read rule, write the predictor as $\hat{z}_{\tau+1} = F(z_\tau, a_\tau, h_\tau)$ (Eq. (3), window suppressed). Training advances $h$ at every position, so $F$ only sees pairs $(z_\tau, h_\tau)$ in which $h_\tau$ summarizes the history up to $\tau$. A rollout that holds $h$ at $h_t$ instead calls $F(\hat{z}_{t+j}, a_{t+j}, h_t)$ for $j = 1, \ldots, m$, pairing an imagined latent $j$ steps ahead with a state $j$ steps stale; training never shows this input, and the mismatch grows with $j$ whenever the hidden variable moves. Advancing $h$ with the imagined $(\hat{z}, a)$ removes the staleness at one carrier step per candidate per imagined step, and the imagination-consistent term addresses the remaining mismatch between imagined and encoded latents ($\text{B}_0$ is trained without it; supplementary material, Section S17). A frozen belief still supplies the plan-time estimate and loses only its update along each imagined trajectory. Freezing costs MemJEPA-B more with the arm hidden than in full view on Reacher, is confounded with rollout damage on PushT, and is not detectable on TwoRoom (Table 12).

# I PROTOCOL DETAILS

This appendix gives the evaluation and training protocol of the LeWM experiments. Success is each environment's native predicate; on PushT the block must lie within 20 px and $\pi/9$ rad of the goal pose, the goal is $d$ steps later in an expert demonstration, and the budget is $2d$. On PushT, occluded arms use our parity re-implementation of LeWM's training ($224 \times 224$ frames), and fully observed waypoint experiments use the released checkpoint; fresh draws (seeds 43–45) exclude training, validation and development episodes. The occluded-PushT construct pairs two real episodes that share the pusher's start pose and the goal but not the block. The block is erased over steps 10 to 84, so the two members' masked frames are bit-identical, and planning starts at step 60 (step 20 in the secondary construct). Each member's own expert continuation succeeds, while the other member's continuation and waiting both fail, so a memoryless planner cannot solve both. Development uses 50 pairs and confirmation 115 pairs; Table 13 lists the held-out sets and the arms evaluated on each.

Training masks come from the simulator. Each clip stays fully visible with probability $p_{\text{full}}$ and is otherwise erased from frame $k_{\text{vis}}$, for every arm and its twin; $p_{\text{full}} = 1$ gives the no-erasure controls. At test the planner receives only the occluded frames and the goal image. Training uses one RTX 4090 per seed and takes 74 to 78 minutes on Reacher for either memory model, or 15 to 45 minutes when only the belief trains on the frozen predictor. Table 15 gives the hyperparameters, and Tables 16, 17 and 18 compare read sites, erasure schedules and belief widths.

Intervals come from a paired bootstrap (10,000 resamples, fixed seed), clustered by episode pair (Reacher), case or episode; tests are exact two-sided sign tests on discordant pairs. The bootstrap pools the three training seeds and omits between-seed variance, so seed-level comparisons are descriptive; per-seed differences between MemJEPA-S, MemJEPA-B and $B_0$ are in the supplementary material (Section S20). The memoryless failure on Reacher is not a tuning artifact: other horizons (3 and 10 blocks) and 20 CEM iterations reach at most 11.0%. Before any memory arm ran we excluded two constructs, because hiding the entity cost the memoryless planner less than the 20 points of Section 4.1. On TwoRoom's two door constructs the base planner scores 14 and 25% with the door visible and 3 and 7% with it hidden (44.0% on the random-door set). On OGBench cube-single (Park et al., 2025) the recipe-matched memoryless planner keeps its success with the UR5e arm erased (40.7% full view, 38.3% hidden), hiding the cube costs at most 7 points, and the released checkpoint scores below its published figure (Table 11). Section S20 of the supplementary material gives the contrasts between the arms of Table 14, pair matching, and agent details for PointMaze, Memory Maze (Pasukonis et al., 2023) and Mortar Mayhem (Pleines et al., 2025).

Table 13: Held-out sets, the arms read on each, and the main reuses of an already-opened set (in reading order; Table 12 lists further controls read on S2).

| Set | First read | Later reads on the opened set |
|---|---|---|
| Reacher held-out (S2) | memoryless, cache, $B_0$, MemJEPA-S | MemJEPA-B; 10-frame window; removal controls |
| Reacher confirmation | frozen read (output site) | advanced read and the advancing ladder (condition C2 varied; App. H) |
| Reacher $L = 20$ | all arms of Table 14 | – |
| PushT pusher held-out | memoryless, cache, $B_0$, MemJEPA-S | MemJEPA-B; 10-frame window; removal controls |
| PushT goal 60 | output-site arms | input-site belief and its partners |
| TwoRoom agent held-out | an earlier variant (own-prediction training) | memoryless, cache, $B_0$, MemJEPA-S; then MemJEPA-B; 10-frame window; removal controls |

# J PLANNING COST

This appendix measures what the belief costs at planning time and whether more planning compute can replace it. Per plan at 300 candidates, the belief takes 1.09× the memoryless planning time in the same run, with 2.6% more FLOPs and 4.95% more parameters; MemJEPA-S takes 0.47×. Compute does not substitute for memory (Table 19, Figure 8): with the arm hidden, the memoryless planner

Table 14: Table 2 with contrasts against the memoryless twin (success %; 95% CIs, Reacher pair-clustered, PushT and TwoRoom case-clustered). Windows: memoryless predictors with 10- or 40-frame context. †Post-hoc comparator on the opened sets. ‡Read after $B_0$ had opened each set (Table 13). §Cache on MemJEPA-B's model. –: not run.

| | Reacher (arm) | | Reacher |
|---|---|---|---|
| Arm | hidden | full view | $L = 20$, hidden |
| Memoryless twin | 7.5 | 93.8 | 8.2 |
| Last-visible cache | 25.3 | – | 5.7§ |
| 10-frame window | 74.0† | 94.7† | 7.7 |
| 40-frame window | 60.0 | 91.8 | 51.8 |
| More compute (1200 candidates) | 9.5 | 96.2 | – |
| MemJEPA-B‡ | 93.0 | **94.7** | 58.2 |
| − memoryless | +85.5 [+82.7, +88.2] | +0.8 [−0.8, +2.5] | +50.0 [+44.7, +55.3] |
| MemJEPA-S | 93.2 | 93.0 | 84.5 |
| − memoryless | +85.7 [+81.7, +89.5] | −0.8 [−2.7, +0.8] | +76.3 [+71.5, +81.0] |

| | PushT (pusher hidden) | | TwoRoom (agent hidden) | |
|---|---|---|---|---|
| Arm | hidden | full view | hidden | full view |
| Memoryless twin | 3.3 | 51.7 | 29.3 | 84.3 |
| Last-visible cache | 15.8 | – | 45.0 | – |
| 10-frame window | 32.5† | 74.7† | 78.0† | 82.7† |
| MemJEPA-B‡ | 52.0 | **78.8** | 79.0 | 83.8 |
| − memoryless | +48.7 | +27.2 | +49.7 | −0.5 |
| 95% CI | [+44.5, +52.8] | [+23.2, +31.3] | [+45.2, +54.2] | [−2.3, +1.3] |
| MemJEPA-S | 57.0 | 72.0 | **83.3** | **87.0** |
| − memoryless | +53.7 | +20.3 | +54.0 | +2.7 |
| 95% CI | [+48.3, +58.8] | [+14.3, +26.3] | [+47.7, +60.2] | [−0.3, +5.7] |

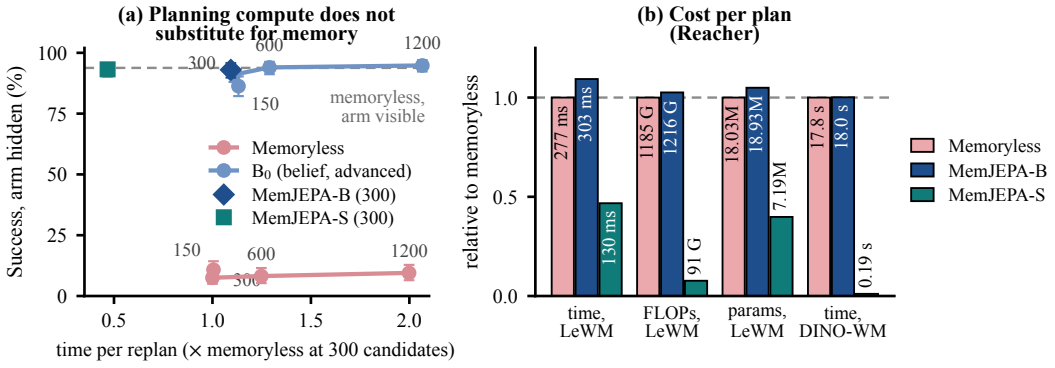



Figure 8: Compute does not substitute for memory. (a) Reacher, held-out: $B_0$ against the memoryless planner over CEM candidates (point labels); MemJEPA-B and MemJEPA-S at 300 candidates; time relative to memoryless at 300. (b) Per-plan cost relative to memoryless, absolutes on the bars.

stays near its floor at four times the candidates, while the belief keeps its gain at half, at no full-view cost. A second timing run on another machine is in the supplementary material (Section S21).

## K Limitations

Context windows catch up past the training clip. A 40-frame window beats $B_0$ at 33 Reacher frames, where MemJEPA-S was not evaluated (Figure 4), and on TwoRoom, with the agent hidden 10 steps before the first decision, the post-hoc 10-frame window matches MemJEPA-B (78.0 against 79.0%, Table 14, interval in the supplementary material, Section S20). A mixed schedule that keeps

Table 15: Training hyperparameters on the LeWM tasks (encoder frozen throughout). $k_{\text{vis}}$: frames visible at the start of an erased clip; $p_{\text{full}}$: probability that a training clip stays fully visible. DINO-WM follows its own training code.

| Hyperparameter | Memoryless | MemJEPA-B | MemJEPA-S |
|---|---|---|---|
| Optimizer | AdamW, weight decay $10^{-3}$ | | |
| Learning rate | $5 \cdot 10^{-5}$ | $5 \cdot 10^{-5}$ | $3 \cdot 10^{-4}$ |
| LR schedule | linear warm-up, then cosine annealing stepped per epoch | | |
| Batch size (clips) | 128 | | |
| Training steps (5 epochs) | Reacher 48,575; PushT 41,705; TwoRoom 13,045 | | |
| Precision, gradient clip | bf16, gradient-norm clip 1.0 | | |
| Clip length (frames) | 11 | | |
| Context window (latents) | 3 | 3 | n/a |
| Frame skip (action block) | 5 primitive steps per latent | | |
| $k_{\text{vis}}$, $p_{\text{full}}$ | 2, 0.5 | | |
| Loss | next-latent MSE | next-latent MSE | MSE + KL[†] |
| Imagination-consistent inputs[‡] | no | $p$=0.5, $j \sim \mathcal{U}\{3..9\}$ | no |
| State carried across clips | n/a | PushT, TwoRoom | no |
| Seeds | 0, 1, 2 | | |
| CEM samples, elites | 300 samples, top 30 | | |
| CEM iterations | Reacher 10; PushT and TwoRoom 30 | | |
| Planning horizon | 5 blocks = 25 steps, whole plan executed before replanning | | |

[†]Decoder MSE to the unmasked latent plus KL (balance 0.8, free bits 1 nat per visible step). [‡]Each clip is selected with probability $p$; from input position $j$ on, the inputs are the model's own multi-step imagined latents (detached).

Table 16: Memory inside the predictor (held-out sets; success %, three seeds pooled). Frozen: the belief held at its last real update inside imagined rollouts; advanced: stepped with the imagined latent and candidate action. PushT recall: goal 60, 115 pairs. –: not run. TwoRoom memoryless: $B_0$'s paired partner (MemJEPA-B's, 29.3%, is in Table 14). Other sets: Section S20.

| Arm | Reacher arm hidden | PushT pusher hidden | PushT recall | TwoRoom agent hidden |
|---|---|---|---|---|
| *hidden variable masked* | | | | |
| No memory (LeWM+CEM, fine-tuned) | 7.5 | 3.3 | 26.8 | 28.5 |
| 40-frame window | 60.0 | – | – | – |
| Last-visible cache | 25.3 | 15.8 | **75.2** | 45.0 |
| Belief inside, frozen | – | – | 17.5 | – |
| Belief inside, advanced, output site | 75.2 | – | – | – |
| $B_0$: belief inside, advanced, input site | **91.2** | **47.7** | 41.3 | **75.5** |
| state swapped between episodes | 6.5 | – | 7.0 (frozen) | – |
| *hidden variable visible* | | | | |
| No memory (LeWM+CEM, fine-tuned) | 93.8 | 51.7 | – | 84.7 |
| Belief inside, advanced, output site | 92.8 | – | – | – |
| $B_0$: belief inside, advanced, input site | 92.8 | 79.2 | – | 82.5 |

standard-onset erasure works on Reacher and TwoRoom, at a 2.3-point full-view cost on TwoRoom, but fails on PushT, and blank-frame training alone does not transfer to an erased arm (Table 17). The belief is a point estimate that the planner never doubts, and neither one-step own-prediction training nor planning to look helps (Section 4.2). The memory was tested on hidden intervals up to 165 steps, where $B_0$ falls to 24.8% (supplementary material, Section S12). The primary contrasts rest on three training seeds per cell, a frozen encoder and three simulated tasks, each seed trained on one RTX 4090 (App. I).

Three established settings failed a precondition before any memory test could run. A memory contrast needs a base latent that encodes the controlled entity, a base planner at full-view parity and planning that succeeds before occlusion, since memory can repair missing history but not an unus-

Table 17: Schedule coverage: the module trained on an equal-thirds per-clip mix of standard onset, random onset and all-black frames, as in Loci-Looped's blackouts (Traub et al., 2024) (success %, case-clustered 95% CIs). [a]MemJEPA-B stack, pair-clustered. [b]Nearest twin (v2). [e]Trained on all-black frames only; notes in the supplementary material, Section S20.

| Task | Standard view, hidden (vs standard belief) | Full view (vs memoryless) | All-black frames (vs recorded twin; vs v2 twin) |
|---|---|---|---|
| Reacher (stack)[a] | 91.8 vs 91.0, $+0.8\,[-1.5,\ +3.2]$ | 93.7 vs 93.8, $-0.2\,[-1.8,\ +1.5]$ | 91.8 vs 6.5, $+85.3\,[+82.5,\ +88.2]$; vs v2 twin 11.0, $+80.8\,[+77.7,\ +83.8]$ |
| TwoRoom | 75.7 vs 73.3, $+2.3\,[-1.3,\ +6.0]$ | 82.0 vs 84.3, $-2.3\,[-4.3,\ -0.3]$ | 73.7 vs 27.5, $+46.2\,[+39.2,\ +53.2]$; vs v2 twin[b] 51.5, $+22.2\,[+14.5,\ +29.8]$ |
| PushT | 20.7 vs 49.8, $-29.2\,[-34.8,\ -23.5]$ (fails the bar) | 62.8 vs 51.7, $+11.2\,[+5.3,\ +17.0]$ ($-17.5$ vs belief's 80.3) | 47.0 vs 2.7, $+44.3\,[+38.7,\ +50.0]$ |
| Reacher, mask-free[e] | 7.0 vs 7.5 memoryless, $-0.5\,[-3.2,\ +2.2]$ (late onset 6.0, $-4.3\,[-7.3,\ -1.3]$) | 94.7 (CI includes 0) | 93.3 vs 6.5, $+86.8\,[+84.0,\ +89.5]$; vs stack $+1.5\,[-0.8,\ +3.8]$ |
| Reweighted (1/2, 1/4, 1/4) | Reacher 92.5; PushT 26.3 ($-23.5\,[-29.0,\ -18.0]$, fails the bar); TwoRoom 79.7 ($+6.3\,[+3.0,\ +9.8]$) | 94.0; 67.3; 83.3 | 91.7; 42.3; 77.7 |

Table 18: Belief width on the frozen-predictor path (only the belief trains; success %, case-clustered 95% CIs). The 0.89M width was inherited from the carrier comparison and never tuned. TwoRoom saturates at 0.89M; on PushT the 8× belief beats the fully fine-tuned belief (47.7% hidden) by $+10.0\,[+4.5,\ +15.3]$.

| | hidden | | | full view | |
|---|---|---|---|---|---|
| Belief | rate | vs memoryless | vs 0.89M | rate | vs 0.89M |
| *TwoRoom* | | | | | |
| 0.17M (h128) | 38.5 | $+9.2\,[+1.8,\ +16.3]$ | $-33.2\,[-39.8,\ -26.3]$ | 81.3 | – |
| 0.89M (h340) | 71.7 | $+42.3$ | – | 81.2 | – |
| about 8× (h1024) | 75.7 | $+46.3\,[+39.8,\ +52.8]$ | $+4.0\,[-0.3,\ +8.3]$ | 81.2 | – |
| *PushT (memoryless 3.3% hidden)* | | | | | |
| 0.17M (h128) | 35.8 | $+32.5\,[+27.2,\ +38.0]$ | $-12.0\,[-16.3,\ -7.5]$ | 67.8 | $-4.7\,[-9.0,\ -0.3]$ |
| 0.89M (h340) | 47.8 | $+44.5$ | – | 72.5 | – |
| about 8× (h1024) | 57.7 | $+54.3\,[+48.7,\ +60.0]$ | $+9.8\,[+5.2,\ +14.3]$ | 77.5 | $+5.0\,[+1.3,\ +8.7]$ |

able representation or an incapable planner, and the three LeWM tasks meet all three. On Searing Spotlights from Memory Gym (Pleines et al., 2025), the frozen encoder does not encode the agent (a position probe errs by 23.9 px against a 6 px bar). On DINO-WM's Wall our base reached 23.3% against the released planner's 96%, failing parity, and on the released checkpoint the trained belief left the prediction loss unchanged. On robomimic Lift and Can the behavior-cloning seed alone beat the planner (64 against 46% on Can) and plain CEM scored 0 of 50 on both, so no memory contrast could be measured (App. G). PLDM is below parity and is reported as internally paired only (App. E). CLWM's Occluded Push (Zhou & Zhang, 2026) is unreleased, and FIRM-WM (Wu et al., 2026) is not released as a planner over a frozen encoder. Ported, it becomes a CEM planner with a recurrent state on LeWM latents, which our standard-trained RSSM row and the no-erasure belief of Table 12 already cover.

An implementation audit (in the supplementary material) found no defect affecting the results. Some numbers recorded before the advanced read was adopted use the frozen read (marked in Tables 16 and 13), and earlier masked PushT rows used a previous mask that also perturbed contact physics (re-

Table 19: Planning compute on Reacher (held-out S2, one machine): memoryless and $B_0$ at 150 to 1200 CEM candidates, all else shared (episode-cluster bootstrap 95% CIs; median ms per replan on an idle RTX 4090).

| CEM candidates | 150 | 300 | 600 | 1200 |
|---|---|---|---|---|
| Memoryless, hidden | 10.8 | 7.5 | 8.2 | 9.5 |
| $B_0$, hidden | 86.3 | 91.0 | 94.0 | 94.8 |
| difference, hidden | $+75.5$ | $+83.5$ | $+85.8$ | $+85.3$ |
| 95% CI | $[+70.2, +80.7]$ | $[+79.7, +87.2]$ | $[+81.8, +89.7]$ | $[+81.7, +89.0]$ |
| Memoryless, full view | 90.2 | 93.8 | 96.2 | 96.2 |
| $B_0$, full view | 91.3 | 92.7 | 96.3 | 96.7 |
| difference, full view | $+1.2$ | $-1.2$ | $+0.2$ | $+0.5$ |
| 95% CI | $[-0.5, +3.0]$ | $[-3.0, +0.7]$ | $[-1.0, +1.3]$ | $[-0.8, +1.8]$ |
| ms per replan, memoryless | 253 | 251 | 314 | 502 |
| ms per replan, $B_0$ | 284 | 279 | 324 | 519 |

run with the render-only mask, the belief changes by $-0.4\,[-2.7,\ +1.9]$ points and the memoryless model is unchanged).

## L THE USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) as assistants throughout this project. They helped search for and summarize related work, suggested and critiqued experiment designs, helped write and debug the training, evaluation and figure code, run experiments and interpret results, and helped draft and edit text, including the write-up of the appendix arguments. The authors chose the research direction and experiments, fixed the held-out case lists and success criteria before evaluation, checked every reported number against per-episode records with a separate re-tally script, verified each reference, and reviewed all text. The authors take full responsibility for the content.