REMEMBER SMARTER: VISUAL HISTORY COMPRESSOR AND HYPERBOLIC EXPERIENCE SPACE FOR ROBOTIC MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot transfer in complex long-horizon manipulation depends on reusing successful experiences acquired while executing simpler subtasks. This setting requires a policy to retain visual context throughout the current trajectory and retrieve relevant experience from other trajectories. However, many VLA models use only a bounded set of recent visual frames and lack an explicit mechanism for reusing successful prior trajectories. Mamba scales linearly with sequence length and is therefore well suited to compressing visual histories, while hyperbolic space can represent hierarchical relations among related experiences. Based on these properties, we propose Remember Smarter (RS), a module that adds a visual history compressor and a hyperbolic experience memory to existing VLA models. The visual branch uses a bidirectional spatial Mamba and a causal temporal Mamba to summarize patch sequences from multiple camera views. Residual cross-attention adds this summary to the VLM states used for action generation while preserving the original sequence of visual tokens. The experience branch encodes final-layer VLM states from successful simple-subtask trajectories into a Poincaré VAE latent space and organizes the resulting codes with a HypHC-inspired binary topology. Retrieved experiences are converted into prompt tokens for action generation. When integrated with π_0, RS improves the overall success rate on LIBERO-Plus by 16.9 percentage points over the base policy. RS also improves real-robot performance, including zero-shot transfer to task compositions that require maintaining object-name grounding, coping with visual occlusion, or recognizing state changes that are visually subtle or unobservable, such as button presses.

1 INTRODUCTION

Vision-Language-Action (VLA) models ground language instructions in visual observations to predict robot actions (Black et al., 2026; Brohan et al., 2023a;b; Kim et al., 2024). However, their memory remains fragile on long-horizon tasks. Complex tasks often require composing multiple simpler subtasks, and recalling successful experiences from these subtasks can improve the reliability of solving the overall task. This calls for experience memory, which stores successful interaction episodes and enables agents to reuse prior experience in new task compositions. Such memory can also maintain object-name grounding across visual and task-context changes. Successful interactions can then be reused when related tasks require the same object in a new composition. Long-horizon manipulation also requires visual memory. Task-relevant object states are often only partially observable. They may be occluded by the robot or other objects. In addition, some interactions leave only subtle or transient visual evidence, such as pressing a button, opening and then closing a door, or tightening a bottle cap. Visual memory therefore maintains a compact representation of prior observations, allowing the agent to recover states that are no longer directly visible and to assess the outcomes of earlier actions.

We propose **Remember Smarter (RS)**, a plug-and-play memory module comprising a visual history compression branch and a hyperbolic experience retrieval branch. For visual memory, RS first applies bidirectional snake-scan Mamba blocks to spatially compress the visual patches from each view. A temporal Mamba block then aggregates the resulting features across the observation history

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.

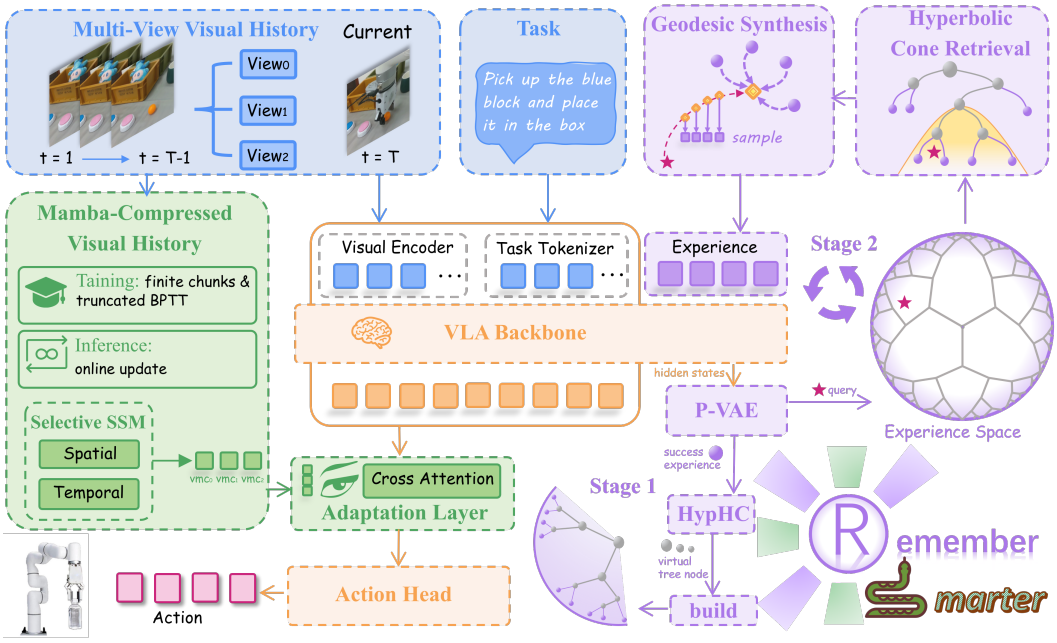


Figure 1: Overview of the Remember Smarter architecture. The visual-history branch compresses multiview patch histories, while the experience branch stores and retrieves reusable experience. Stage 1 denotes offline construction of the fixed experience snapshot from successful source episodes (illustrated by simple subtasks) and its Poincaré memory space. Stage 2 denotes online retrieval for a target, potentially complex task and injection of experience tokens into the VLM.

and produces a compact representation for each view (Gu & Dao, 2024; Dao & Gu, 2024). Consecutive training fragments pass recurrent state across their boundaries while stopping gradients every 16 frames; at inference, the recurrent state is updated online. When using the visual compression module, an additional adapter is required to interface it with the action head of different VLA models. For experience memory, RS retains only successful training episodes and stores the final VLM representation used for action prediction from each episode. A Poincaré variational autoencoder (P-VAE) provides latent coordinates for encoding experiences during storage and for representing queries during retrieval. During experience storage, an auxiliary copy of the leaves at a common radius is optimized using a compact set of triplets inspired by Hyperbolic Hierarchical Clustering (HypHC) (Chami et al., 2020). The optimized representations are then decoded to yield a binary tree topology. During experience retrieval, an asynchronous beam search retrieves up to eight memory leaves and identifies their lowest common ancestor (LCA). The LCA representation in the Poincaré latent space guides four geodesic prompt tokens inserted into the language model prompt.

Our contributions are summarized as follows:

- We propose **Remember Smarter (RS)**, a plug-and-play memory architecture that equips VLA policies with a visual history memory and a hierarchical experience memory while preserving the VLA backbone architecture.
- We introduce a visual history compressor that uses spatial and temporal Mamba. During training, it applies truncated backpropagation through time (TBPTT) with state passing. During online inference, it performs causal streaming with fixed states.
- We introduce a hierarchical hyperbolic experience space that incorporates retrieval coordinates from a Poincaré variational autoencoder (P-VAE), binary tree construction inspired by HypHC, entailment cone retrieval, and geodesic experience token generation.

Figure 1 summarizes the two memory paths and their insertion points in the VLA action pathway.

2 Related Work

2.1 Challenges and Emerging Directions in VLA

Vision-Language-Action (VLA) models unify visual perception, language grounding, and action generation for robot control (Brohan et al., 2023b;a; Kim et al., 2024; Black et al., 2026; Intelligence et al., 2025). Generalist robot policies increasingly leverage heterogeneous demonstrations and shared model interfaces across tasks and embodiments (Collaboration et al., 2025; Walke et al., 2024; Khazatsky et al., 2025). Deploying them in long-horizon settings remains difficult: a policy must retain task state, recover from failed or ambiguous interactions, and do so within the latency budget of closed-loop control. These requirements have motivated lightweight VLA backbones (Shukor et al., 2025; Lin et al., 2025; Luo et al., 2026; Wang et al., 2025), memory-augmented policies (Shi et al., 2026b; Li et al., 2025a; Torne et al., 2026; Sridhar et al., 2025; Fang et al., 2025), and hierarchical or long-horizon control methods (Fan et al., 2025; Li et al., 2025b; Liu et al., 2025). Action parameterization is a separate issue. Flow-matching VLA policies can model continuous action distributions (Black et al., 2026), whereas 6D rotation representations avoid the discontinuities associated with Euler-angle parameterizations (Zhou et al., 2020). RS addresses neither the action parameterization nor planning; it adds memory to an existing VLA action pathway.

The data and evaluation ecosystem also spans task-prompted manipulation benchmarks (Jiang et al., 2023; Mees et al., 2022), broad multi-task suites (Yu et al., 2021; James et al., 2019; Gu et al., 2023), and robustness-oriented bimanual benchmarks (Mu et al., 2025; Chen et al., 2025). These resources establish diverse sources of policy experience, but they do not by themselves specify how a deployed policy should select a small, relevant subset of successful prior episodes at control time.

2.2 External and Episodic Memory for Control

Robot memory systems have combined episodic and semantic memories for task planning (Nasir et al., 2019), organized life-long robot experience hierarchically (Bärmann et al., 2025), and modeled object-centric memory-state progression (Chung et al., 2025). Recent VLA memory systems instead expose visual or language-conditioned experience to a robot policy (Shi et al., 2026b; Li et al., 2025a; Torne et al., 2026; Sridhar et al., 2025; Fang et al., 2025; Zhou et al., 2026; Li et al., 2026). RoboMME makes the importance of evaluating such memory mechanisms explicit for robot generalist policies (Dai et al., 2026). RS is complementary: it keeps the within-episode state causal and compact, whereas cross-episode records are fixed successful demonstrations retrieved asynchronously for the VLM.

2.3 State Space Models and Mamba

Long visual histories are difficult to handle in Transformer-based VLA policies. Dense self-attention evaluates interactions among temporal patch tokens, so a multi-view context becomes increasingly expensive as its length grows. Selective state-space models, including Mamba (Gu & Dao, 2024; Dao & Gu, 2024), instead update a compact latent state recurrently. Their input-dependent parameterization selects which information to retain while processing a sequence in linear time.

Existing work uses SSMs either to preserve temporal dynamics before reducing video tokens (Jiang et al., 2025; Kim & Seo, 2026) or as part of the VLA policy itself (Liu et al., 2024). Studies of state capacity, length extrapolation, and model compression further identify the need to retain useful information within bounded SSM state (Chen et al., 2026; Wang, 2024; Ezoe & Sato, 2024; He et al., 2024). RS instead retains the pretrained VLA backbone and applies Mamba only to compress each view's patch history into a fixed-size representation that the original action pathway can read.

2.4 Hyperbolic Geometry for Hierarchical Representation

Hyperbolic space has exponentially increasing volume with radius and is widely used for latent hierarchies and taxonomies (Nickel & Kiela, 2017; 2018). Hyperbolic graph models provide a concrete construction for learning in this geometry (Chami et al., 2019), including geometry-aware aggregation through Fréchet means (Lou et al., 2021). Poincaré VAEs provide probabilistic hyperbolic latent coordinates (Mathieu et al., 2019); HypHC decodes hierarchies from hyperbolic embeddings

(Chami et al., 2020); and entailment cones express directed ancestor–descendant relations (Ganea et al., 2018). RS combines these tools for experience organization and retrieval. The learned topology is used only for routing and prototype construction, not as a source of task labels. Unlike a single undifferentiated history buffer, RS separates a causal, action-side multiview state for within-episode control from an immutable, successful-episode VLM library for cross-episode reuse; asynchronous retrieval ensures that the latter never stalls the action loop.

3 PRELIMINARIES

3.1 PROBLEM SETTING

We study long horizon robotic manipulation with a pretrained VLA backbone. At time t, the robot receives a language instruction L, a multiview image set $I_t \in \mathbb{R}^{V\times H\times W\times 3}$, where V is the number of camera views, and the preceding visual history $I_{<t} = \{I_1, \ldots, I_{t-1}\}$. The policy predicts

$$a_t = \pi_\theta(I_t, L, M_t, \mathcal{E}_t), \tag{1}$$

where M_t is compact visual memory and $\mathcal{E}_t$ denotes reusable experience from prior demonstrations. For manipulation, a_t typically specifies the Cartesian pose of the robot end effector, optionally together with a gripper state.

3.2 STATE SPACE MODELS FOR VISUAL HISTORY

State space models maintain a fixed size hidden state while processing a sequence. Given an input x_t, a discrete selective SSM updates its state and emits an output as

$$h_t = \overline{\mathbf{A}}_t h_{t-1} + \overline{\mathbf{B}}_t x_t, \qquad y_t = \mathbf{C}_t h_t, \tag{2}$$

where h_t is the hidden state. In Mamba, the discretized transition $\overline{\mathbf{A}}_t$, input projection $\overline{\mathbf{B}}_t$, and output projection $\mathbf{C}_t$ depend on the current input through a selective parameterization (Gu & Dao, 2024; Dao & Gu, 2024). This recurrence processes sequences in linear time and retains historical information in a fixed size state. Equation 2 makes explicit the fixed-state recurrence that RS uses for temporal, rather than backbone, processing.

We use this mechanism to encode the visual history rather than to replace the VLA backbone. Let E_{vis} be the shared visual encoder and let $\mathcal{C}_\phi$ denote the visual history compressor, which applies spatial processing within each view and causal temporal SSM updates across frames. The resulting memory is

$$M_t = \mathcal{C}_\phi\left(\{E_{\text{vis}}(I_\tau^v)\}_{\substack{\tau=1,\ldots,t\\ v=1,\ldots,V}}\right). \tag{3}$$

Thus, M_t summarizes the multiview observations through the current frame in a fixed-size representation that can be read by the action prediction pathway. The current image is also retained in the original VLA visual-token stream. At inference, the compressor updates this representation recurrently without retaining raw visual history, as defined in Eq. 3.

The visual history branch is trained through the policy supervision; it uses no separate reconstruction or distillation objective. For a demonstration with target actions $a_t^\star$, its training objective is

$$\mathcal{L}_{\text{VMC}} = \frac{1}{T}\sum_{t=1}^{T} \ell_{\text{act}}(\pi_\theta(I_t, L, M_t, \mathcal{E}_t), a_t^\star), \tag{4}$$

where ℓ_{act} is the action prediction loss used by the VLA policy; Eq. 4 is optimized with the same supervision as the underlying policy. Implementation details for state passing during training and the history length study are deferred to the appendix.

3.3 HIERARCHICAL HYPERBOLIC EXPERIENCE SPACE

The experience branch organizes successful final-layer VLM representations for reuse at inference. Let r_i be the hidden state at the last valid prefix token of successful demonstration i. A Poincare

variational autoencoder (PVAE) maps it to a retrieval coordinate $z_i = \mu_\phi(r_i) \in \mathbb{B}^d$ (Mathieu et al., 2019), and is trained with the standard reconstruction and regularization objective

$$\mathcal{L}_{\text{PVAE}} = \sum_i \left[\ell_{\text{rec}}(r_i, D_\omega(z_i)) + \beta\, \text{KL}(q_\phi(z \mid r_i) \,\|\, p(z))\right]. \tag{5}$$

The resulting coordinates $Z = \{z_i\}$ are retained as the retrieval space; Eq. 5 is optimized before the immutable memory snapshot is used for retrieval.

To expose hierarchical structure without altering Z, we learn an auxiliary set of routing coordinates Y and decode a binary topology $\mathcal{T}$ from them. Internal prototypes P are then calibrated to represent directed ancestor and descendant relations through entailment cones:

$$\mathcal{T} = D_{\text{tree}}\left(\arg\min_Y \mathcal{L}_{\text{tri}}(Y; Z)\right), \qquad P^\star = \arg\min_P \mathcal{L}_{\text{cone}}(P; \mathcal{T}, Z). \tag{6}$$

Here, D_{tree} is a deterministic tree decoding operator that maps the optimized routing coordinates to a binary topology. It has no learnable parameters and is distinct from D_ω, the Poincare VAE decoder used to reconstruct VLM representations. $\mathcal{L}_{\text{tri}}$ preserves the proximity structure of the Poincare coordinates over sampled triples, whereas $\mathcal{L}_{\text{cone}}$ calibrates directed containment relations implied by the decoded topology. The auxiliary optimization does not modify the retrieval coordinates Z. The hierarchy objective uses a compact triplet relaxation inspired by HypHC and cone calibration for directed relations (Chami et al., 2020; Ganea et al., 2018); Eq. 6 therefore separates representation learning, topology decoding, and prototype calibration.

At time t, the final VLM hidden state at the learned [MEM_QUERY] token is encoded as a query q_t for a subsequent action step. At action time, $\mathcal{E}_t$ instead contains the latest valid result published from an earlier query, or masked empty slots if none is available. For a valid result, the hierarchy retrieves relevant leaves, obtains the prototype p_a of their lowest common ancestor, and generates reusable experience tokens by interpolating from that result's query toward p_a along the corresponding hyperbolic geodesic. Thus, q_t can affect only a later experience input, preserving the non-blocking action contract in Eq. 1. Metric definitions, interpolation details, and retrieval implementation are deferred to the appendix.

4 METHODS

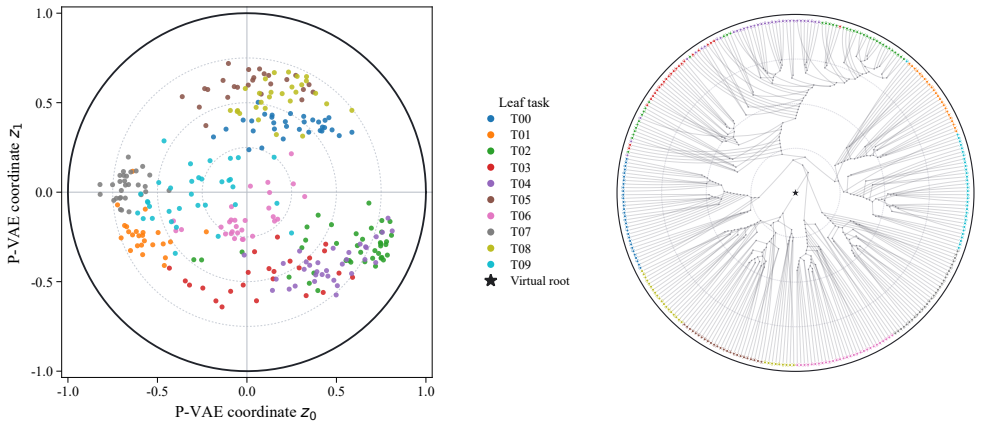
4.1 OVERVIEW

RS augments a pre-trained VLA policy with two complementary memory paths (Figure 1). The visual-history compressor retains recent multi-view changes in a fixed-size recurrent state. The experience memory stores successful demonstrations as a hierarchy and retrieves reusable patterns without extending the VLA context with raw history. The two paths are lightweight add-ons: they preserve the original visual-token stream and use the original action interface rather than introducing a new action head. We first describe the online visual-state path, then the offline experience snapshot and its asynchronous use at action time.

4.2 MAMBA-COMPRESSED VISUAL HISTORY

At each decision step, the compressor reuses frozen VLA-encoder features of the current images together with its cached preceding features; the current image also remains in the original VLA visual-token stream. A lightweight adapter maps each view's patch features to a shared width. Within a frame, bidirectional spatial Mamba scans the patch grid in a serpentine order; a learned query then pools the resulting patch features into one vector per view. Two causal temporal Mamba layers update a separate recurrent state for each view, while sharing parameters across views. Thus, the memory captures both within-frame spatial context and chronological visual changes, but its size does not grow with episode length.

The current frame is the final temporal element, so its per-view memory vectors condition the current action through a single residual cross-attention adapter: final action-facing VLA states provide the queries, and the visual memory provides keys and values. The enhanced states are passed to the original action head. We train the adapter and compressor only through the policy loss, passing the



(a) P-VAE experience space **(b) HypHC hierarchy**

Figure 2: **P-VAE experience space and HypHC-inspired hierarchy.** (a) Successful experiences are represented by 16-dimensional deterministic P-VAE posterior locations. For visualization only, their origin-log-mapped tangent coordinates are projected by PCA to two dimensions and mapped back to a two-dimensional Poincaré disk; the displayed axes are not original retrieval dimensions. Colored points denote leaf tasks. Under the guidance of decoded internal nodes, task experiences appear in distinct regions in this visualization. (b) The corresponding decoded binary hierarchy, with the virtual root indicated by a star.

temporal state between truncated BPTT fragments. The resulting recurrent cache is retained for the next decision, so neither past images nor patch tokens need to be retained. Backbone-specific feature paths, masking, and training settings are given in Appendix B.1.

4.3 Hyperbolic Experience Space

Storage and organization. After the policy-adaptation stage, RS builds an immutable memory snapshot from the final-layer hidden state of the last valid prefix token in successful demonstrations only, with exactly one record per eligible source episode. A Poincaré variational autoencoder (P-VAE) (Mathieu et al., 2019) maps each state to a hyperbolic retrieval coordinate; its deterministic posterior location is stored as a memory leaf, and its decoder maps selected latent points back to the VLM-representation space. During policy adaptation, experience slots are masked because this snapshot has not yet been constructed; they are enabled only after the offline memory is fixed. Thus, experience retrieval supplies inference-time conditioning without creating a second policy-supervision path. RS then learns a HypHC-inspired binary topology over an auxiliary copy of these leaves (Chami et al., 2020). Keeping this optimization separate preserves the P-VAE coordinates used for retrieval. Each internal node is represented by the Karcher mean of its descendants and calibrated with an entailment cone, which supplies a directed parent-to-descendant routing signal in addition to hyperbolic distance (Karcher, 1977; Ganea et al., 2018). Formal definitions and optimization details are deferred to Appendix B.2. Retrieval and topology construction operate on the original 16-dimensional P-VAE coordinates. Figure 2 visualizes them only after a log-map–PCA–exponential-map transformation into a two-dimensional disk: the decoded internal nodes guide a coarse-to-fine organization in which experiences from different tasks occupy apparently distinct regions in the visualization. The fixed visualization sample and transformation are specified in Appendix C.2.

Asynchronous retrieval and token composition. At time t, the final VLM hidden state of a learned [MEM_QUERY] token is encoded as a query in the same P-VAE space for a subsequent action step. A background worker traverses the binary hierarchy, using hyperbolic distance and internal-node entailment cones to prioritize relevant subtrees before ranking at most eight leaves. The action loop never waits for this computation. A valid published result yields the lowest common internal ancestor, a shared coarse experience prototype, and four equally budgeted points along the

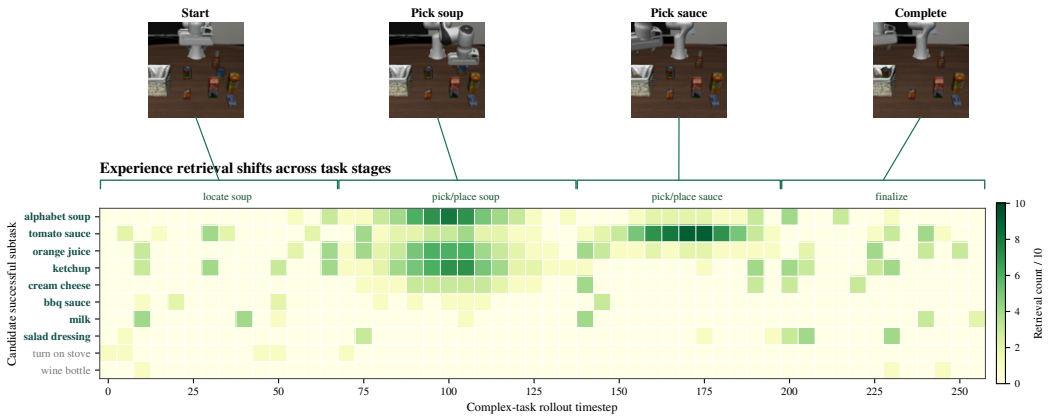


Figure 3: **Stage-aware experience retrieval trace.** The top row shows successive states of a complex manipulation task. For each rollout timestep, a heatmap cell gives the number of ten rollouts in which a candidate simple-task record is retrieved (color bar, count$/10$). The trace is diffuse at task initiation and completion, but concentrates on the object-relevant records during the two manipulation stages.

hyperbolic geodesic from the submitted query toward that prototype; these points are decoded as experience tokens. At time t, only the latest valid result from a prior query is inserted immediately before the first image embedding; otherwise its four slots are masked. A learned [MEM_QUERY] token after the task prompt forms the next query. The original VLM self-attention therefore fuses any available experience with the visual and language context without an additional fusion module. The queue, validity checks, and exact-retrieval fallback are specified in Appendix B.2.

Figure 3 reports an aggregated retrieval trace of this stage-aware behavior. Retrieval can remain diffuse at task initiation and completion, while the action stages concentrate on successful simple-task experiences that are relevant to the object currently being manipulated.

5 Experiments

We evaluate RS only with π_0 in three settings: simulation on LIBERO-Plus, real-robot experiments, and component analysis of visual compression and hyperbolic experience retrieval on standard LIBERO. The experiments answer three questions: whether RS improves zero-shot robustness under controlled perturbations, whether the integrated module helps on memory-oriented physical tasks, and whether the visual and experience paths make separable contributions.

5.1 Simulation Benchmarks

Following LIBERO-Plus (Fei et al., 2025), we evaluate robustness over seven perturbation dimensions: camera viewpoint, robot initial state, language instruction, lighting, background, sensor noise, and object layout. Both π_0 and π_0+RS start from the official π_0–LIBERO checkpoint: the foundation model is pre-trained on large-scale robot data and fine-tuned on the original LIBERO demonstrations. RS is adapted only with those original LIBERO demonstrations; it is not trained on LIBERO-Plus trajectories. We evaluate each of the 10,030 LIBERO-Plus task instances once. Table 1 reports strict success rates over the full benchmark. RS raises total success from 53.6% to 70.6%, a 16.9-point increase. Published results for the other methods are retained in Table 1 as contextual reference only; we make no cross-paper ranking claim because their reported protocols need not coincide with ours.

5.2 Real-Robot Memory Tasks

We evaluate π_0 and π_0+RS across seven real-robot task conditions; their definitions, trial counts, and strict-success criterion are given in Appendix C.5. The suite tests four memory-centric dimensions: repeated-action counting (Press-2/4/6), ordered event sequences (Order), object–name alignment

Table 1: Robustness on the full LIBERO-Plus benchmark. Entries are strict success rates (%). Total is computed over all 10,030 tasks rather than averaging the seven dimension-level scores. Published baselines are reproduced for contextual reference and retain the settings reported in their original work; they are not controlled head-to-head comparisons with π_0+RS. The MemoryVLA variants are evaluated in the zero-shot setting (Shi et al., 2026a). The final row is the absolute change of π_0+RS relative to π_0, computed from unrounded benchmark totals; it can therefore differ slightly from subtracting the displayed rounded values.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
MemoryVLA++	36.4	68.9	88.7	93.8	90.6	63.5	83.8	73.1
MemoryVLA	42.7	44.9	84.4	92.8	95.0	62.1	84.7	70.2
OpenVLA-OFT	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
π_0-Fast	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0+RS	50.8	45.8	65.3	92.1	94.0	81.2	78.0	70.6
Δ over π_0	↑37.0	↑39.8	↑6.5	↑7.1	↑12.6	↑2.2	↑9.1	↑16.9

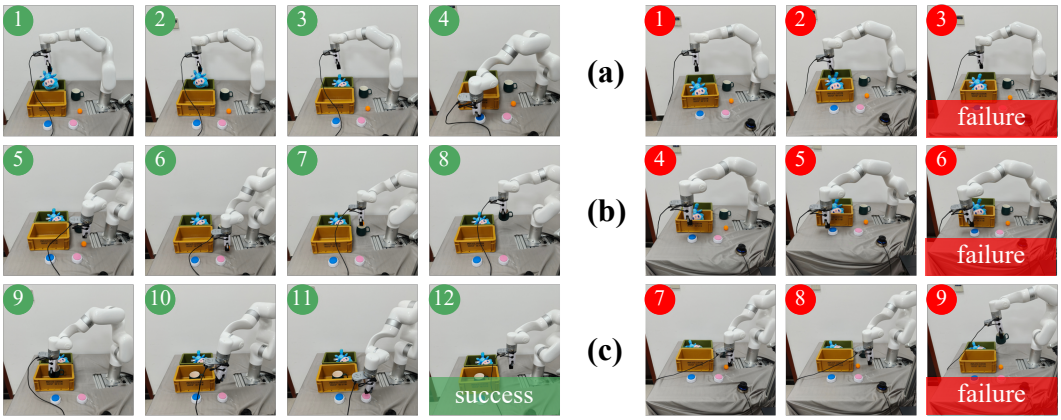


Figure 4: Representative real-robot rollouts on the compositional Chen.-H task. The green sequence shows a successful π_0+RS rollout. The red sequences show π_0 failures: (a) failure to lift Chenchen, (b) failure to press the required button, and (c) failure to place the ball in the cup. These qualitative examples illustrate distinct failure modes; they do not isolate the causal contribution of any individual RS component.

(Chenchen), and compositional long-horizon execution (Chen.-H). Occluded placement provides an additional perceptual stressor. We highlight the compositional *Chenchen-hard* (Chen.-H) task in the main text: *First, move Chenchen from the yellow box to the green box. Then press the blue button once. Next, put the yellow ball into the green cup, and place the green cup into the yellow box. Finally, press the pink button once.* The task requires retaining object identity and recognizing the completion status of multiple subgoals. RS is trained only on the simpler conditions and receives no additional Chen.-H training.

The right panel of Figure 5 reports higher strict success for RS in six of the seven conditions. On held-out Chen.-H, success rises from 20% for π_0 to 80% for π_0+RS. The only decrease is on short-horizon Chen.-S (64% to 60%); the largest increase is on Chen.-H (+60 points).

Figure 4 provides rollout-level context for the Chen.-H result and makes the strict outcome labels visually auditable.

5.3 Ablations and Analysis

Component contributions. We compare the base π_0, π_0+VMC, and full π_0+RS on the four standard LIBERO suites (Liu et al., 2023). VMC contains only the visual Mamba compressor; RS ad-

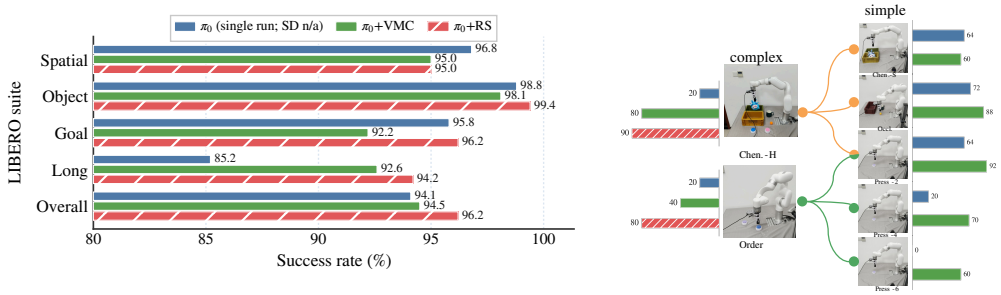


Figure 5: Component ablation on the four standard LIBERO suites (left) and real-robot success by task condition (right). VMC is the visual Mamba compressor without the Poincaré experience space. π_0 is a single run (no variability estimate); π_0+VMC and π_0+RS bars are five-run means. The fixed figure does not display error bars. On the right, links identify the simpler training conditions associated with the evaluation conditions; the bars report strict success.

ditionally enables the Poincaré experience space. As shown in Figure 5, VMC raises Long success from 85.2% to 92.6% in the displayed ablation. Adding experience memory further raises Long to 94.2% and the four-suite overall score to 96.2%, a 2.1-point increase over π_0. This pattern is concentrated on the long-horizon suite: RS preserves comparable performance on the short Spatial, Object, and Goal suites, with its largest improvement on Long.

6 Conclusion

We have presented Remember Smarter (RS), a memory module that augments VLA policies with visual-history compression and hierarchical experience retrieval. The visual branch reuses the VLA visual encoder, applies bidirectional spatial and causal temporal Mamba, and feeds a fixed-size per-view history through a residual reader into the action-facing states. The experience branch maps successful final-layer VLM representations to Poincaré VAE leaves, builds a separate HypHC-inspired routing topology, and uses a non-blocking retrieval worker to form four geodesic experience tokens.

On the full LIBERO-Plus benchmark, RS raises total success from 53.6% to 70.6%. It also improves strict success on six of seven real-robot conditions and improves the LIBERO-Long component ablation. These results are consistent with the use of compact online visual history and asynchronously retrieved successful experience for the long-horizon settings evaluated here. The evidence is limited to the π_0 instantiation and to memory built from successful source episodes. In addition, asynchronous retrieval avoids blocking the action loop but has no sublinear worst-case guarantee. Evaluating other VLA backbones and characterizing sensitivity to source-memory availability are important next steps.

AI Assistance Statement

Large language models and other generative AI tools were used to aid drafting and polish the manuscript; identify and retrieve related work; assist research ideation and execution; draft portions of the research text; and assist in checking mathematical derivations and claims. The authors reviewed all AI-assisted suggestions, verified citations, methods, mathematical claims, and reported results against the underlying sources and experiments, and take responsibility for the final content. Generative AI tools were not used to generate synthetic datasets or replace experimental evaluation.

REFERENCES

Leonard Bärmann, Chad DeChant, Joana Plewnia, Fabian Peller-Konrad, Daniel Bauer, Tamim Asfour, and Alex Waibel. Episodic memory verbalization using hierarchical representations of life-long robot experience. In *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, pp. 783–790. IEEE, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023a. URL https://arxiv.org/abs/2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023b. URL https://arxiv.org/abs/2212.06817.

Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks, 2019. URL https://arxiv.org/abs/1910.12933.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering, 2020. URL https://arxiv.org/abs/2010.00402.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025. URL https://arxiv.org/abs/2506.18088.

Yingfa Chen, Xinrong Zhang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Stuffed mamba: Oversized states lead to the inability to forget, 2026. URL https://arxiv.org/abs/2410.07145.

Nhat Chung, Taisei Hanyu, Toan Nguyen, Huy Le, Frederick Bumgarner, Duy Minh Ho Nguyen, Khoa Vo, Kashu Yamazaki, Chase Rainwater, Tung Kieu, Anh Nguyen, and Ngan Le. Rethinking progression of memory state in robotic manipulation: An object-centric perspective, 2025. URL https://arxiv.org/abs/2511.11478.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui

Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies, 2026. URL https://arxiv.org/abs/2603.04639.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024. URL https://arxiv.org/abs/2405.21060.

Haruka Ezoe and Kazuhiro Sato. Model compression method for s4 with diagonal state space layers using balanced truncation, 2024. URL https://arxiv.org/abs/2402.15993.

Yiguo Fan, Pengxiang Ding, Shuanghao Bai, Xinyang Tong, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, Zhaoxin Fan, Badong Chen, and Donglin Wang. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation, 2025. URL https://arxiv.org/abs/2508.19958.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation, 2025. URL https://arxiv.org/abs/2501.18564.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings, 2018. URL https://arxiv.org/abs/1804.01882.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023. URL https://arxiv.org/abs/2302.04659.

Wei He, Kai Han, Yehui Tang, Chengcheng Wang, Yujie Yang, Tianyu Guo, and Yunhe Wang. Densemamba: State space models with dense hidden connection for efficient large language models, 2024. URL https://arxiv.org/abs/2403.00818.

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

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment, 2019. URL https://arxiv.org/abs/1909.12271.

Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, Sungjin Ahn, Jan Kautz, Hongxu Yin, Yao Lu, Song Han, and Wonmin Byeon. Storm: Token-efficient long video understanding for multimodal llms, 2025. URL https://arxiv.org/abs/2503.04130.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023. URL https://arxiv.org/abs/2210.03094.

Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

Geewook Kim and Minjoon Seo. State-space hierarchical compression with gated attention and learnable sampling for hour-long video understanding in large multimodal models, 2026. URL https://arxiv.org/abs/2506.13564.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Runhao Li, Wenkai Guo, Zhenyu Wu, Changyuan Wang, Haoyuan Deng, Zhenyu Weng, Yap-Peng Tan, and Ziwei Wang. Map-vla: Memory-augmented prompting for vision-language-action model in robotic manipulation, 2025a. URL https://arxiv.org/abs/2511.09516.

Wenhao Li, Xiu Su, Dan Niu, Yichao Cao, Hongyan Xu, Zhe Qu, Lei Fan, Shan You, and Chang Xu. Sentinel-vla: A metacognitive vla model with active status monitoring for dynamic reasoning and error recovery, 2026. URL https://arxiv.org/abs/2605.01191.

Ying Li, Xiaobao Wei, Xiaowei Chi, Yuming Li, Zhongyu Zhao, Hao Wang, Ningning Ma, Ming Lu, and Shanghang Zhang. Manipdreamer: Boosting robotic manipulation world model with action tree and visual guidance, 2025b. URL https://arxiv.org/abs/2504.16464.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment, 2025. URL https://arxiv.org/abs/2511.04555.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation, 2024. URL https://arxiv.org/abs/2406.04339.

Ruizhe Liu, Pei Zhou, Qian Luo, Li Sun, Jun Cen, Yibing Song, and Yanchao Yang. Himacon: Discovering hierarchical manipulation concepts from unlabeled multi-modal data, 2025. URL https://arxiv.org/abs/2510.11321.

Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean, 2021. URL https://arxiv.org/abs/2003.00335.

Yuankai Luo, Woping Chen, Tong Liang, Baiqiao Wang, and Zhenguo Li. Simvla: A simple vla baseline for robotic manipulation, 2026. URL https://arxiv.org/abs/2602.18224.

Emile Mathieu, Charline Le Lan, Chris J. Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders, 2019. URL https://arxiv.org/abs/1901.06033.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022. URL https://arxiv.org/abs/2112.03227.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins, 2025. URL https://arxiv.org/abs/2504.13059.

Jauwairia Nasir, Deok-Hwa Kim, and Jong-Hwan Kim. Art neural network-based integration of episodic memory and semantic memory for task planning for robots. *Autonomous Robots*, 43(8):2163–2182, 2019.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations, 2017. URL https://arxiv.org/abs/1705.08039.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry, 2018. URL https://arxiv.org/abs/1806.03417.

Physical Intelligence. openpi: Open-source models and packages for robot learning. https://github.com/Physical-Intelligence/openpi, 2026. Software repository, accessed 2026-08-05.

Hao Shi, Weiye Li, Bin Xie, Yulin Wang, Renping Zhou, Tiancai Wang, Xiangyu Zhang, Ping Luo, and Gao Huang. Memoryvla++: Temporal modeling via memory and imagination in vision-language-action models, 2026a. URL https://arxiv.org/abs/2606.09827.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation, 2026b. URL https://arxiv.org/abs/2508.19236.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Memer: Scaling up memory for robot control via experience retrieval, 2025. URL https://arxiv.org/abs/2510.20328.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. Mem: Multi-scale embodied memory for vision language action models, 2026. URL https://arxiv.org/abs/2603.03596.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

Shida Wang. Longssm: On the length extension of state-space models in language modelling, 2024. URL https://arxiv.org/abs/2406.02080.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model, 2025. URL https://arxiv.org/abs/2509.09372.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. URL https://arxiv.org/abs/1910.10897.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks, 2020. URL https://arxiv.org/abs/1812.07035.

Yijing Zhou, Qiwei Liang, Sitong Zhuang, Jiaxi Li, Xianpeng Wang, Boyang Cai, Yunyang Mo, and Renjing Xu. Action-effect memory pretraining for robot manipulation, 2026. URL https://arxiv.org/abs/2606.12499.

A Mathematical Derivations

This appendix states the masked recurrence and residual-reader operators used by RS. Let $\tilde{s}_\tau^v$ and $\tilde{m}_\tau^v$ be the state and output proposed by the two-layer temporal Mamba module for view v. With availability indicator $c_\tau^v \in \{0, 1\}$, the implemented update is

$$s_\tau^v = c_\tau^v \tilde{s}_\tau^v + (1 - c_\tau^v) s_{\tau-1}^v, \qquad m_\tau^v = c_\tau^v \tilde{m}_\tau^v + (1 - c_\tau^v) m_{\tau-1}^v. \tag{7}$$

Equation 7 shows that an unavailable camera neither changes its recurrent cache nor overwrites its most recent memory token. For one attention head with valid-view mask c_t^v, the history reader computes

$$a_{ij} = \frac{\exp(q_i^\top k_j/\sqrt{d_h}) c_t^{v_j}}{\sum_{r=1}^{V} \exp(q_i^\top k_r/\sqrt{d_h}) c_t^{v_r}}, \qquad \widetilde{h}_i = h_i + \gamma W_O \sum_{j=1}^{V} a_{ij} v_j, \tag{8}$$

where all attention weights are defined as zero when no view is valid. The multi-head implementation applies Eq. 8 to each head and concatenates the resulting contexts. Finally, the four retrieved experience tokens are points on the Poincare geodesic

$$\Gamma_{q,p}(\tau) = \mathrm{Exp}_q\big(\tau \,\mathrm{Log}_q(p)\big), \qquad \tau \in \{0.45, 0.55, 0.65, 0.75\}, \tag{9}$$

Equation 9 defines the four fixed token locations from the current query q to the prototype p of the retrieved lowest common ancestor. These points are decoded independently before entering the original VLM self-attention stack.

B Detailed Framework Description

Unless stated otherwise, D_{SigLIP}, D_{VLM}, and D_A denote the hidden widths of the frozen visual encoder, the VLM, and the action expert, respectively. The remaining fixed widths used by RS are stated explicitly where they are introduced.

B.1 Implementation Details of Mamba-Compressed Visual History

Backbone feature paths and adapters. The history pathway reuses the frozen π_0 visual encoder (Black et al., 2026) through the shared openpi policy interface (Physical Intelligence, 2026). The frozen SigLIP (Zhai et al., 2023) forward pass returns 16×16 patches through `pre_logits_2d`. After stop-gradient, an independently trained $\mathrm{LayerNorm}(D_{\mathrm{SigLIP}})+\mathrm{Linear}(D_{\mathrm{SigLIP}}, 256)$ adapter maps these patches to the compressor width. Original VLM image tokens are unchanged. No PCA or auxiliary distillation objective is used. Training caches contain complete episode patch sequences and view masks, but never Mamba memories or recurrent states.

Compressor configuration and view masking. The compressor width is $d = 256$, with state size 16, convolution width 4, expansion factor 2, and Mamba head width 64. One Mamba layer operates in each spatial direction; aligned outputs are averaged, pooled by one learned query, and passed through a learned projection. Two causal temporal Mamba layers then update per-view histories. Temporal parameters are shared across the ordered multi-view stream, whose fixed positions are front, left, and right, whereas convolution and SSM states remain separate for each view. For availability mask $c_\tau^v \in \{0, 1\}$, the temporal update is

$$(m_\tau^v, s_\tau^v) = \begin{cases} \mathrm{Mamba}^{(2)}_{\mathrm{temp}}(p_\tau^v, s_{\tau-1}^v), & c_\tau^v = 1, \\ (m_{\tau-1}^v, s_{\tau-1}^v), & c_\tau^v = 0, \end{cases} \tag{10}$$

so an unavailable view changes neither its recurrent state nor its latest memory representation, exactly as specified by Eq. 10.

Residual reader and backbone seams. The history reader uses four attention heads and residual scale $\gamma = 0.05$. Its three memory tokens have width 256, while query, key, value, and output projections use the pinned action-expert width D_A; each head therefore has width $D_A/4$ rather than a backbone-independent fixed width. Memory masks remove unavailable views from the attention logits, and an item with no valid memory receives an exactly zero residual update. Disabling visual history bypasses the reader and preserves the original action path. The action-facing VLA states read by the adapter are instantiated in π_0 by `suffix_out` before `action_out_proj` at every flow-matching step. This provides a single residual history-reading interface without replacing the π_0 action expert or the independently trained RS modules.

Online state contract. At inference, each sampled frame performs one spatial encoding and one mask-conditional temporal update for every available view. The update follows the current frame and supplies M_t to the current action computation; its state is then retained as history for the next decision. Episode reset clears convolution caches, SSM states, latest view memories, and the sampling timer, so no state crosses episode boundaries. Recurrent-state shape is independent of episode length. Appendix D gives TBPTT and optimization details.

B.2 Formalization of the Hyperbolic Experience Space

Poincare geometry and numerical convention. We use curvature magnitude $c = 1$ and the open ball $\mathbb{B}^d$. For floating point tensors, $\epsilon = 10^{-7}$ for float32/bfloat16/float16 and 10^{-12} for float64. The implementation clamps `artanh` inputs to $[-1+\epsilon, 1-\epsilon]$ and projects points to radius $1 - 10\epsilon$. With $\lambda_x = 2/(1 - \|x\|^2)$ and Mobius addition $\oplus$, the common operations are

$$d_{\mathbb{B}}(x, y) = 2\,\mathrm{artanh}(\| - x \oplus y\|), \tag{11}$$

$$\mathrm{Exp}_x(v) = x \oplus \left(\tanh\frac{\lambda_x\|v\|}{2}\frac{v}{\|v\|}\right), \qquad \mathrm{Log}_x(y) = \frac{2\,\mathrm{artanh}(\| - x \oplus y\|)}{\lambda_x\| - x \oplus y\|}(-x \oplus y). \tag{12}$$

Equation 11 supplies the distance used by routing, and Eq. 12 supplies the exponential and logarithmic maps used for geodesic interpolation and Karcher updates. Norms and denominators are clamped by ϵ, with continuous zero limits. A Karcher solve starts from the projected weighted Euclidean mean and runs 12 updates $m \leftarrow \mathrm{Exp}_m(0.5 \sum_i w_i \mathrm{Log}_m z_i)$.

P-VAE leaves and query. For every eligible final-layer hidden state $h_i \in \mathbb{R}^{D_{\mathrm{VLM}}}$, the wrapped-normal P-VAE uses

$$(\mu_i, \ell_i) = f_{\mathrm{enc}}(h_i), \qquad z_i = \mathrm{Exp}_0(0.25 \tanh \mu_i), \qquad \sigma_i = \mathrm{softplus}(\mathrm{clamp}(\ell_i, -8, 4)) + 10^{-4}, \tag{13}$$

The encoder first maps h_i through $\tanh(\mathrm{Linear}(D_{\mathrm{VLM}}, 128))$ and uses separate projections to produce the 16-dimensional μ_i and $\ell_i = \log \sigma_i^2$. Its decoder maps $\mathrm{Log}_0(\Pi(z)) \in \mathbb{R}^{16}$ through $\tanh(\mathrm{Linear}(16, 128))$ and $\mathrm{Linear}(128, D_{\mathrm{VLM}})$. The objective is reconstruction MSE plus $\beta = 10^{-3}$ times the KL term. Training samples in the origin tangent space as $0.25 \tanh \mu_i + \sigma_i \epsilon$ before applying Exp_0. Equation 13 defines the deterministic retrieval location used at inference. Sampling is used in training, whereas retrieval always uses projected posterior locations. The successful-episode filter is applied before the immutable snapshot is built. The online [MEM_QUERY] hidden follows the same deterministic $z = \mathrm{Exp}_0(0.25 \tanh \mu_\phi(h))$ path to obtain q_t; it has no additional head or cone semantics.

Compact HypHC-inspired triples and tree decode. Let $Z = (z_1, \ldots, z_N)$ be P-VAE leaves and let $\rho = 0.75$. An independent auxiliary copy is initialized as

$$y_i = \Pi\left(\rho \frac{z_i}{\max(\|z_i\|, 10^{-6})}\right), \tag{14}$$

where Π is the Poincare-ball projection. Gradients through this phase do not modify Z. We form $s_{ij} = \exp[-d^2_{\mathbb{B}}(z_i, z_j)/(2b^2)]$, where b is the median nonzero pairwise P-VAE distance, and sample distinct triples (a, b, c). For the three pairs $\mathcal{P} = \{(a, b), (a, c), (b, c)\}$, the implemented compact loss is

$$w_{ij} = \frac{\exp[-d_{\mathbb{B}}(y_i, y_j)]}{\sum_{(r,s)\in\mathcal{P}} \exp[-d_{\mathbb{B}}(y_r, y_s)]}, \qquad \mathcal{L}_{\mathrm{tri}} = \mathbb{E}\left[\left(\sum_{(i,j)\in\mathcal{P}} s_{ij} - \sum_{(i,j)\in\mathcal{P}} w_{ij} s_{ij}\right)^2\right]. \tag{15}$$

After each update, y_i is reprojected to radius ρ, subject to the numerical floor in Eq. 14. This is a differentiable triplet relaxation in Eq. 15, inspired by HypHC rather than the full HypHC training system.

All auxiliary pairs (y_i, y_j) are sorted by descending dot product, with stable leaf-ID ties. A Union–Find single-link pass merges disconnected components until the deterministic binary tree has $2N - 1$ nodes. The decoded root serves as a virtual router. For retrieval-space initialization, each internal prototype is recomputed as the Karcher mean of its descendant P-VAE leaves z_i; equal-radius y_i values remain auxiliary.

Topology-only cone calibration. For $K = 0.1$, a non-root internal prototype p has half-aperture and exclusion radius

$$\psi(p) = \arcsin\left(K \frac{1 - \|p\|^2}{\|p\|}\right), \qquad r_{\min} = \frac{2K}{1 + \sqrt{1 + 4K^2}}. \tag{16}$$

The angle is computed between $-\mathrm{Log}_p(0)$ and $\mathrm{Log}_p(x)$, with cosine clamped to $[-1 + 10^{-7}, 1 - 10^{-7}]$, and $E_{\mathrm{cone}}(p, x) = [\angle_p(x) - \psi(p)]_+$. If $\|p\| < r_{\min}$, this energy is marked invalid and routing uses distance only, as required by Eq. 16.

Topology supplies positives from non-root internal-to-child edges and internal-to-descendant-leaf closure. For each positive parent, outside-subtree leaves with the lowest initial cone energy supply hard negatives. With an adaptive radial margin m_{px}, calibration minimizes

$$\begin{aligned}\mathcal{L}_{\mathrm{cal}} = {} & \lambda_+ \mathbb{E}_{(p,x)\in\mathcal{P}} E_{\mathrm{cone}}(p, x) + \lambda_- \mathbb{E}_{(p,n)\in\mathcal{N}}[\gamma - E_{\mathrm{cone}}(p, n)]_+ \\ & + \lambda_r \mathbb{E}_{(p,x)\in\mathcal{P}}[\|p\| + m_{px} - \|x\|]_+ + \lambda_a \mathbb{E}_p d^2_{\mathbb{B}}(p, p^{(0)}).\end{aligned} \tag{17}$$

Relations are generated solely from the decoded topology and never read task, subtask, or action labels. Optimization variables are tangent coordinates for non-root internal prototypes only. P-VAE leaves and the virtual root are copied back exactly after optimization. The four terms in Eq. 17 respectively enforce positive containment, negative separation, radial ordering, and proximity to the Karcher initialization.

Beam and exact retrieval. For a non-root internal node v, the routing score is

$$s_{\text{node}}(v \mid q_t) = 0.25\, d_{\mathbb{B}}(p_v, q_t) + E_{\text{cone}}(p_v, q_t). \tag{18}$$

Equation 18 combines distance with cone compatibility. The approximate search begins at the virtual root's children, retains the eight lowest-scoring internal nodes at each level, and expands their children. Visited leaves accumulate until the usable pool reaches $\max(8, \lceil 2 \cdot 8 \rceil) = 16$. A visited leaf ℓ is ranked by

$$s_{\text{leaf}}(\ell \mid q_t) = d_{\mathbb{B}}(q_t, z_\ell) + \frac{1}{|\mathcal{A}_\ell^{\text{valid}}|} \sum_{v \in \mathcal{A}_\ell^{\text{valid}}} E_{\text{cone}}(p_v, q_t), \tag{19}$$

where the second term is zero if no ancestor has a valid cone. The final ranking in Eq. 19 is applied after excluding records whose episode ID equals the query episode, the best eight leaves are returned. If exclusions leave fewer than the required number, an explicit exact safety completion ranks all leaves.

The exact oracle batches all non-root internal cone energies, propagates ancestor sums and valid counts with one root-to-leaf DFS, batches all leaf distances, and performs the same final top-8 sort. Its per-query cost is $\mathcal{O}(N)$. The beam records visited and pruned nodes and its fallback reason, but has no sublinear worst-case guarantee and may also cost $\mathcal{O}(N)$.

Composer and fixed output. The composer finds the lowest common internal ancestor a_t of the returned top-8 leaves in the decoded topology and obtains its calibrated P-VAE-space prototype p_{a_t}. For zero-based slot j, the implementation uses

$$(\tau_0, \tau_1, \tau_2, \tau_3) = (0.45, 0.55, 0.65, 0.75), \qquad u_j = \text{Exp}_{q_t}(\tau_j \text{Log}_{q_t} p_{a_t}). \tag{20}$$

Equation 20 fixes the interpolation schedule. Decoding $u_0, \ldots, u_3$ yields exactly $4 \times D_{\text{VLM}}$ values. Empty candidates return four zero decoder-width slots to the non-blocking caller but are classified as empty retrieval and never published by the asynchronous worker.

Asynchronous validity. One global capacity-one request queue uses latest-wins submission. Every submission replaces any queued request, increments a sequence number, and records the latest tuple (g, s, e, t) of generation, sequence, episode, and step. A completed worker result is published only when its tuple still matches the latest request and elapsed wall time is within the configured timeout. Empty results, exceptions, and post-completion timeouts increment telemetry and are discarded. The action loop never waits. Poll validity is

$$\text{valid}(\mathcal{R}_\tau, t) = \mathbb{I}[e_\tau = e_t]\, \mathbb{I}[g_\tau = g_t]\, \mathbb{I}[1 \le t - \tau \le 2]. \tag{21}$$

Thus, an age-zero result cannot affect the current action step. Reset increments the generation, clears episode results and step state, and drains the queue. Search also excludes the query episode before ranking.

Prompt attention contract. For each batch row, four projected $\langle \text{exp} \rangle$ embeddings are inserted immediately before the first real image embedding, and one learned [MEM_QUERY] embedding is inserted at the end of the task prompt. Position IDs are recomputed from the valid-token mask, and decoder-only adapters use the corresponding causal or prefix-block attention edges. The existing VLM processes this augmented prompt with its original self-attention. From the final hidden sequence, [MEM_QUERY] is detached and submitted for use only at $t + 1$ or later; original token positions are gathered back before the action expert.

No production experience cross-attention or second residual gate exists. With `enabled=false`, no virtual tokens are inserted and inputs, attention mask, and original action path are returned unchanged. With experience enabled before memory has been populated, masked empty experience slots and a valid [MEM_QUERY] are still inserted so that a future query can be produced.

C Experimental Configuration and Additional Results

C.1 Complete Evaluation Protocol

For controlled simulation comparisons, evaluation instances and initial states are fixed before either policy is run. LIBERO-Plus uses all 10,030 single-perturbation instances described in Section 5.

The reported primary setting trains only on original LIBERO; the released LIBERO-Plus training set is not used. The standard LIBERO component study uses each suite's demonstrations for the corresponding Spatial, Object, Goal, or Long evaluation.

With the visual encoder and PaliGemma prefix frozen, the training configuration updates the SigLIP dimension adapter, bidirectional spatial Mamba blocks, learned-query pooling and projection, two temporal Mamba layers, the residual history reader, and the action expert in Stage B. After this policy-adaptation stage, the P-VAE, independent topology-learning coordinates, and internal-prototype calibration are trained or constructed offline from the successful source records to form an immutable experience library. This library does not enter the VLA optimizer, policy loss, or policy checkpoint. No query-radius head, experience cross-attention, or second residual gate is used.

Memory snapshot and leakage controls. The experience snapshot is constructed after the policy-adaptation stage from successful source demonstrations, with one final-timestep record per successful episode. Each record retains its P-VAE leaf, decoded VLM-space representation, and episode identifier. Retrieval excludes a record from the query episode before ranking, and the asynchronous publication rule in Eq. 21 rejects stale, empty, timed-out, and cross-episode results. Consequently, a future action can consume only a valid result produced for the same episode; the action loop neither waits for search nor receives a fresh result at the step that issued its query.

Storage and search cost. For N source records, the snapshot contains N 16-dimensional P-VAE leaves, their decoded D_{VLM}-dimensional VLM-space representations, and a binary topology with $2N - 1$ nodes. The resulting memory is therefore linear in N and retains no raw images or past patch-token sequences. During ordinary retrieval, the beam retains at most eight internal nodes at each level and returns at most eight leaves. The exact safety completion remains linear in N, as stated after Eq. 19; it is used only when the beam cannot produce the required usable set.

Retrieval-trace reporting. Figure 3 aggregates ten rollouts of one held-out multi-stage task. At each rollout timestep, a candidate record contributes one count when it appears in the returned top-eight set. The plot is therefore a qualitative trace of which records were selected, not an accuracy, relevance, or causal-attribution metric.

C.2 P-VAE and Hierarchy Visualization Protocol

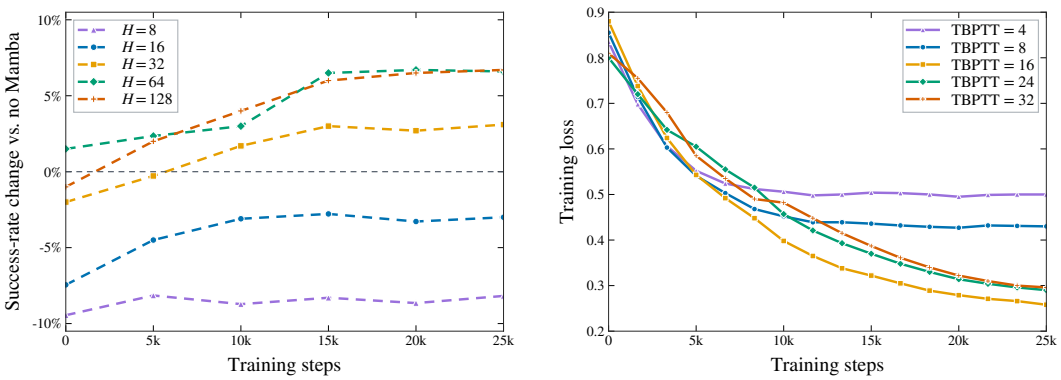
Figure 2 uses the first 300 successful π_0 inference records from the original LIBERO training corpus, spanning the ten tasks denoted T00–T09 in the figure. This sample is used only to make the fixed two-dimensional graphic legible; online retrieval always operates in $\mathbb{B}^{16}$. Specifically, we map the 16-dimensional leaves and decoded internal prototypes to the tangent space at the origin with Log_0, fit PCA in that tangent space, and map the resulting two coordinates to the display disk with Exp_0. Consequently, the apparent two-dimensional separation is illustrative and is not used to compute retrieval distances, tree topology, or success metrics.

C.3 Mamba Training-Window and Fixed-State Streaming Evaluation

For the history-length study, we train VMC with Mamba histories of $H \in \{8, 16, 32, 64, 128\}$ and evaluate the LIBERO-Long success-rate change relative to a no-Mamba baseline every 5k steps. The TBPTT study fixes $H = 64$ and varies the backpropagation truncation length over $\{4, 8, 16, 24, 32\}$. Training loss is logged throughout optimization. Figure 6(a) reports the success-rate change as the history window varies, while Figure 6(b) reports the corresponding TBPTT loss curves for the fixed 64-frame history. The 64-frame configuration was selected as the default because its final success-rate change matches the 128-frame setting in the plotted study while using a smaller recurrent training window. Among the displayed truncation lengths, 16 reaches the lowest final training loss; this curve is an optimization diagnostic rather than a claim of a statistically significant performance difference.

C.4 Component Ablation Details

The component ablation uses only π_0 and compares the base policy, π_0+VMC, and full π_0+RS. Each variant is trained and evaluated under the standard protocol for LIBERO-Spatial, LIBERO-Object,



(a) LIBERO-Long success-rate change

(b) Training loss under TBPTT (H = 64)

Figure 6: History-window and TBPTT sensitivity. (a) LIBERO-Long success-rate change relative to a no-Mamba baseline, measured every 5k training steps for different visual-history windows. (b) Training loss under several TBPTT truncation lengths with the history fixed at 64 frames.

LIBERO-Goal, and LIBERO-Long. No variant changes the demonstrations, data order, optimization budget, checkpoint-selection rule, action parameterization, or evaluation instances within a suite. The grouped plot reports each suite and their unweighted Overall mean. The base π_0 result is a single run; VMC and RS report five-run means. The fixed main-paper figure does not encode cross-run dispersion, so this comparison is not used for a significance claim.

C.5 Real-Robot Task Definitions and Success Criteria

The real-robot suite probes four memory-centric requirements: counting repeated events, preserving an ordered action sequence, grounding a persistent object name, and composing these abilities over a long horizon. Occluded placement is an additional perceptual stressor. The task instructions are as follows:

- **Press-2**: *Press the purple button twice.* Press-4 and Press-6 use the same button with the requested count changed to four and six.
- **Order (Press-hard)**: *First, press the blue button once. Then press the purple button twice. Finally, press the pink button three times.*
- **Occl.**: *First, put the yellow ball into the green cup, then place the green cup into the red box.*
- **Chen.-S**: *Move Chenchen from the yellow box to the green box.* Here, *Chenchen* always denotes the designated plush toy rather than a distractor.
- **Chen.-H**: *First, move Chenchen from the yellow box to the green box. Then press the blue button once. Next, put the yellow ball into the green cup, and place the green cup into the yellow box. Finally, press the pink button once.*

Strict success requires completion of every requested subgoal in the specified order. Any wrong object, wrong order, extra button press, safety stop, timeout, or human intervention is a failure.

Figure 7 shows representative successful keyframe sequences for these task families; it is illustrative and does not replace the strict trial-level evaluation reported in the main text.

Training and generalization conditions. Training uses Press-2, Ordered Button Pressing, Occluded Placement, and the Chenchen-simple condition. Press-4, Press-6, and Chenchen-hard are evaluated without additional task-specific training.

Each method is evaluated on 25 trials for Press-2, Occl., and Chen.-S, and on 10 trials for Press-4, Press-6, Order, and Chen.-H. The corresponding strict success-rate denominators therefore explain

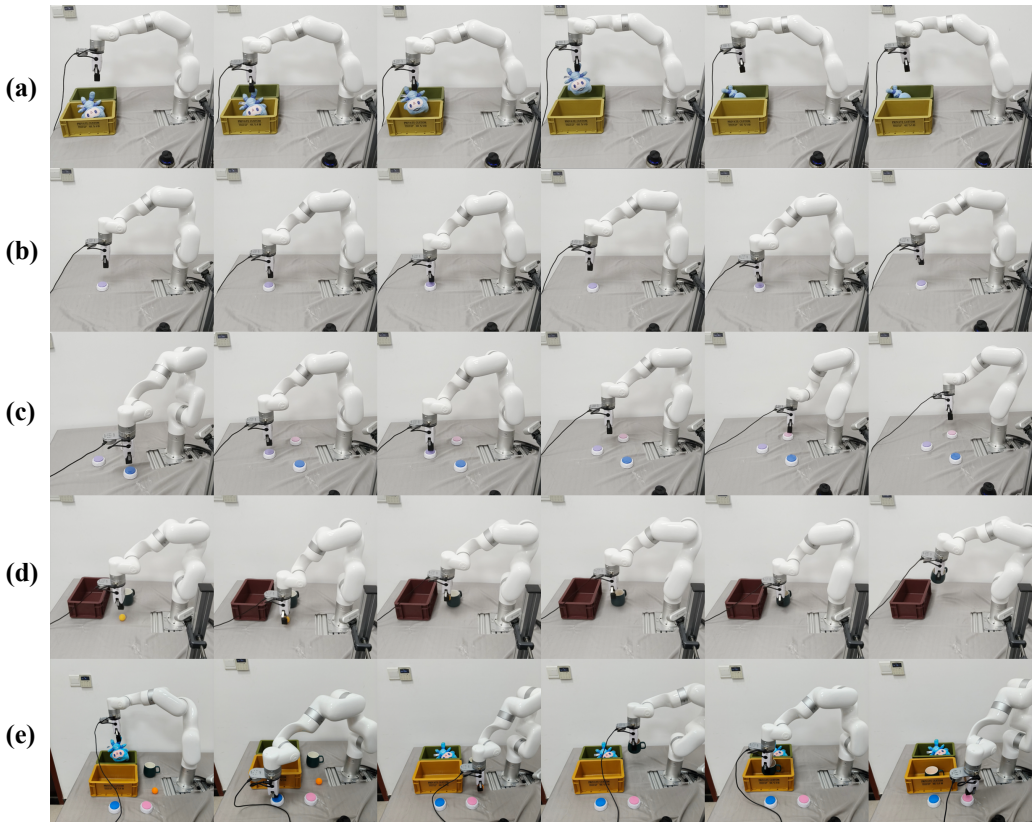


Figure 7: Representative successful keyframe sequences for the real-robot task families: (a) Chen.-S name grounding, (b) Press-2 counting, (c) Order sequence following, (d) occluded placement, and (e) Chen.-H compositional long-horizon execution.

the percentages displayed in the right panel of Figure 5; results are reported as empirical trial proportions, without a significance-test claim. In addition to strict success rate, logging records button-count error, the first sequence error, redundant actions after hidden placement, completed-subgoal fraction, and the first failure stage.

D Training Configuration

RS is trained through the pinned openpi π_0 interface. The released π_0 policy checkpoint initializes the policy, while the visual compressor starts from the seed-0 `mamba_base`; no additional standalone Mamba pretraining is used. Cached inputs contain frozen SigLIP `pre_logits_2d` patch features and view masks, rather than cached recurrent activations. Each example therefore reconstructs its causal history using the current RS parameters.

Table 2 lists the default settings used for the reported π_0+RS configuration.

Each nonzero learning-rate group uses a 100-step warm-up followed by cosine decay to one tenth of its peak rate. The default evaluation uses a 64-frame history. The history-length and TBPTT studies reported in Figure 6 vary those values separately; they are not part of the default training setting.

Table 2: Default π_0+RS training configuration.

Item	Setting
Visual history	3 views, 1 FPS, 64 frames; TBPTT window 16
VMC	$d = 256$, state 16, convolution 4, expansion 2, Mamba head width 64
Precision / parallelism	bf16 activations; fp32 parameters and gradients; 4-device FSDP
Optimizer	Grouped AdamW; weight decay 10^{-4}; gradient clip 1; EMA 0.999
Batch and checkpoints	Global batch 32; checkpoint every 2,500 steps
Stage A	5k steps; RS adapter, VMC, and history reader trainable
Stage B	25k steps; RS policy path and original action expert trainable; PaliGemma prefix frozen
Learning rates	Stage A: 2×10^{-5} (Mamba), 10^{-4} (adapter/reader); Stage B: 10^{-5} (Mamba/adapter), 5×10^{-5} (reader/action)