LaMem-VLA: Dual Latent Memory in Vision-Language-Action Models for Robotic Manipulation

Hongyu Qu[1], **Jianzhe Gao**[2], **Xiaobin Hu**[3], **Shaohuan Yang**[1], **Xinlei Yu**[3], **Rui Yan**[1], **Wenguan Wang**[2], **Xiangbo Shu**[1], **Shuicheng Yan**[3]
[1]Nanjing University of Science and Technology [2]Zhejiang University
[3] National University of Singapore

arXiv:2607.07608v1 [cs.RO] 8 Jul 2026

Abstract

Mainstream Vision-Language-Action (VLA) models predict actions primarily from the current observation under a Markovian assumption, thus struggling with long-horizon, temporally dependent tasks. Existing memory-augmented VLAs either expand the observation window or retrieve history from the memory bank as auxiliary policy-side context. However, they leave memory outside the native latent embedding space of VLA reasoning, preventing historical experience from being fluidly interleaved with multimodal reasoning and action formation. To this end, we introduce **LaMem-VLA**, a latent-memory-native framework that reconstructs historical experience into latent memory tokens and directly interweaves them with VLA reasoning. At its core, **LaMem-VLA** introduces four coordinated components: ***(i)*** a **curator** that organizes historical experience into two complementary short-term and long-term memory vaults; ***(ii)*** a **seeker** that queries both vaults using the multimodal cognition to retrieve context-relevant evidence; ***(iii)*** a **condenser** that reconstructs the retrieved evidence into compact short-term and long-term latent memory tokens; and ***(iv)*** a **weaver** that injects these memory tokens with the current observation and instruction into one continuous embedding sequence. By representing, retrieving, and consuming historical experience entirely in the same continuous latent space, **LaMem-VLA** enables memory to directly participate in VLA reasoning and guide action generation under a bounded context. Extensive experiments on SimplerEnv and LIBERO demonstrate the superiority of our **LaMem-VLA**. The project page will be available at **LaMem-VLA**.

1 Introduction

Vision-language-action (VLA) models [1, 2, 3, 4] have become a promising paradigm for general robotic manipulation. By combining the powerful capabilities of pretrained vision-language models [5, 6, 7] with policy learning [8, 9, 10] on robotic data [11, 12, 13, 14], they map visual observations and language instructions into executable action chunks. Despite this progress, most existing VLA models [2, 1, 15] implicitly rely on a Markovian assumption, predicting actions primarily from the current observation without considering temporal dependencies. This simplification creates a *temporal short-horizon bias*: VLA models can react to the currently visible state, but fail to reason about previous state

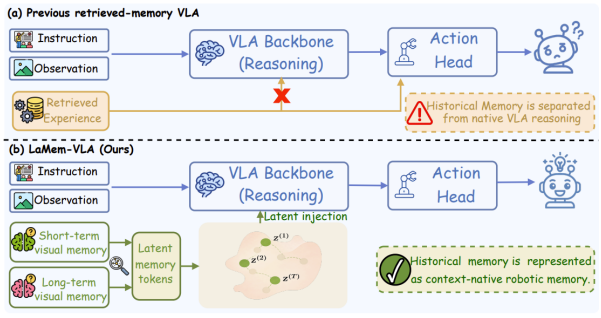

Figure 1: Paradigm comparison of memory-augmented VLA Models. (a) Unlike previous VLA models that store historical experience in an auxiliary memory bank and consume retrieved memory as external policy-side context, (b) **LaMem-VLA** treats historical experience as context-native latent memory, which is stored, retrieved, and consumed in the model embedding space.

transitions, completed operation steps, and the current phase of a multi-step task. As a result, these models especially struggle with long-horizon manipulation tasks.

Recent efforts have sought to alleviate temporal short-horizon bias by augmenting VLA models [8, 16, 17] with historical context or memory mechanisms along two main axes. ***(i)*** One line of work incorporates short-horizon episode context by concatenating historical frames [18, 19] or extending the input into a video sequence [20, 21, 22]. Although such designs expose recent state changes, they incur computational and memory costs that grow with the context length, while the fixed temporal horizon imposes an inherent memory ceiling, causing potentially task-relevant evidence outside the window to be discarded. ***(ii)*** Another line of work [23, 24, 25] retrieves past trajectories or relevant historical tokens from an external memory bank to condition downstream action policies. Although these methods demonstrate the value of historical experience for long-horizon manipulation, they still suffer from an architectural limitation: the historical memory is stored outside the model's native token space and consumed as auxiliary policy-side context after the VLA model reasoning. This rigid separation prevents memory from being fluidly interleaved with the internal reasoning where the VLA model jointly perceives the scene, interprets the instruction, and resolves action queries before action formation.

This limitation raises a more fundamental question for memory-dependent VLA models: *whether historical experience can be represented as context-native robotic memory, stored, retrieved, and consumed in the same continuous space where the VLA model already perceives, reasons, and acts?* Latent embedding space [26, 27, 28, 29, 30, 31, 32] offers a natural answer to this question. Modern VLA models already integrate visual observations and language instructions in a continuous token embedding space [33, 2, 1, 15]; therefore, robotic historical memory can be organized as machine-native latent memory tokens that are compatible with the internal reasoning process. Under this formulation, robotic memory becomes part of the model's operating context rather than an auxiliary scaffold attached after multimodal reasoning. Long-horizon robotic manipulation also calls for two complementary forms of historical memory: **short-term memory** is visually dominant, preserving visually grounded evidence from the current episode, such as object locations and subtle state changes; **long-term memory** is semantically dominant, preserving task progress, contextual semantics, and action continuity across longer horizons. Notably, their distinction lies in *provenance and function* rather than in representation form: both are ultimately reconstructed as latent memory tokens that can be consumed by the VLA model in the latent space. This leads to our pivotal research question:

> 🔍 *How can we architect historical memory as a generative latent faculty, capable of fluidly reconstructing short-term visual evidence and long-term semantic evidence into compact memory tokens that interweave seamlessly with the VLA reasoning and action generation process?*

To answer this question, we propose **LaMem-VLA**, a novel native latent memory framework for robotic manipulation, which explicitly organizes robotic history into two complementary memory vaults, and weaves dual-scale memory into the model reasoning for memory-augmented action generation. At its core, **LaMem-VLA** closes the loop between latent memory weaving and action reasoning through four coordinated modules: First, ♣ a **latent memory curator** factorizes past robotic experience into two complementary vaults: a short-term memory vault for recent visual evidence and a long-term memory vault for semantic and action-continuity evidence. Second, during the action reasoning process, ♦ a **latent memory seeker** builds a context-aware query from the current multimodal cognition state (*i.e.*, the visual and instruction tokens), and uses it to retrieve task-relevant historical evidence from dual memory vaults, grounding the present decision in past perceptual evidence and long-horizon semantic-action continuity. Third, ♥ a **latent memory condenser** compresses these potentially redundant retrieved evidence into fixed-length short-term and long-term latent memory tokens that are compatible with the VLA embedding space. Finally, ♠ a **latent memory weaver** stitches these condensed memory tokens directly into the action reasoning sequence before action query tokens are resolved, allowing historical memory to participate in the same latent reasoning process as the current image, instruction, and action queries. The resulting memory-grounded action queries condition a diffusion-based action expert [8, 34] to generate temporally aware robotic action sequences.

We conduct comprehensive evaluations across two simulators (*i.e.*, LIBERO [35] and SimplerEnv-Bridge [36]). On LIBERO, **LaMem-VLA** reaches an average success rate of **97.6**%, outperforming MemoryVLA [23] by **1.1** points and our baseline CogACT [15] by **4.4** points, while improving over

π_0 [1] by **3.5** points on the first four suites. On SimplerEnv-Bridge, **LaMem-VLA** achieves **73.9**% average success, surpassing our baseline CogACT [15] by **16.6** points and π_0 [1] by **4.7** points. These results indicate that weaving dual-scale latent robotic memory into VLA reasoning boosts the robustness of VLA models beyond policy-side memory conditioning, especially when action generation depends on task progress and historical cues. One limitation of the current version is that the empirical validation is conducted in simulated environments. We are currently extending **LaMem-VLA** to real-world robot platforms, and the corresponding real-world experiments will be included in the next version.

In summary, our main contributions are as follows:

- We introduce a new paradigm for robotic memory in VLA models: historical experience is treated as context-native latent memory, which is stored, retrieved, and consumed in the model embedding space, supporting scene perception, instruction understanding, and action-intent formation within the same latent reasoning process.
- We propose **LaMem-VLA**, a dual latent memory framework for robotic manipulation, which explicitly organizes robotic history into two complementary memory vaults, and weaves dual-scale memory into the model reasoning for memory-augmented action generation.
- We design a latent memory condensing mechanism, which transforms retrieved historical evidence into fixed-length latent short-term and long-term memory tokens that are compatible with the VLA model embedding space.

2 Related Work

Vision-Language-Action (VLA) Models. Driven by advances in pretrained vision-language models (VLMs), VLA models [37, 38, 1, 39] have demonstrated promising performance in robotic control via mapping visual observations and language instructions to robot actions. According to their action policy design, VLA models can be broadly classified into two paradigms: single-stream architecture and hierarchical architectures. **(i)** Single-stream models [40, 41, 42, 43] directly discretize continuous actions [33, 2] within a unified vision-language backbone, and autoregressively predict action tokens in a language-like manner. To meet the high control-frequency requirements in robotic manipulation, many works focus on inference efficiency optimization, *e.g.*, parallel or speculative decoding [44, 45], dynamic LLM layer activation [46], and parameter quantization [47]. **(ii)** Hierarchical models [3, 48, 49] typically employ VLMs for high-level reasoning and planning, and adopt a separate generative policy, such as diffusion-based [1, 15] and flow-matching-based [50, 9] policies, to synthesize smooth and high-quality action trajectories. This separation reduces response latency and facilitates smooth real-world deployment, promoting robust high-level planning alongside high-frequency control for robotic manipulation. Despite these advances, these methods still predict actions primarily from the current observation and instruction without leveraging extended historical context, limiting their robustness in long-horizon manipulation tasks. In contrast, our work aims to equip VLA models with explicit progress awareness via memory mechanisms, addressing memory-dependent tasks that current approaches fail to handle.

Memory Mechanisms for Robotic Control. Mainstream VLA policies are formulated under a Markovian assumption, predicting actions primarily from the current observation and therefore lacking explicit awareness of task progress. Existing attempts to incorporate historical context can be broadly categorized into three groups. **(i)** *Temporal-context expansion:* This line of work directly exposes the policy to temporally extended observations by interleaving historical frames with language tokens [18, 19] or aggregating multi-frame features [20, 21, 22] within a predefined context window. Although straightforward, these methods incur computational and memory costs that grow with the context length, while the fixed temporal horizon imposes an inherent memory ceiling, causing potentially task-relevant evidence outside the window to be discarded. **(ii)** *Sparse history abstraction:* Another paradigm compresses past interactions into lightweight proxy representations, such as recurrent latent states [51, 52], motion-centric cues [53], action summaries [54], or a sparse set of representative observations [55]. By avoiding direct processing of the full observation history, this paradigm improves temporal awareness with relatively low computational overhead. However, aggressive abstraction of historical experience can remove fine-grained perceptual details and high-level semantic context that are critical for long-horizon manipulation tasks. **(iii)** *External memory conditioning:* A third direction [23, 24, 25] stores historical observations and action tokens in an

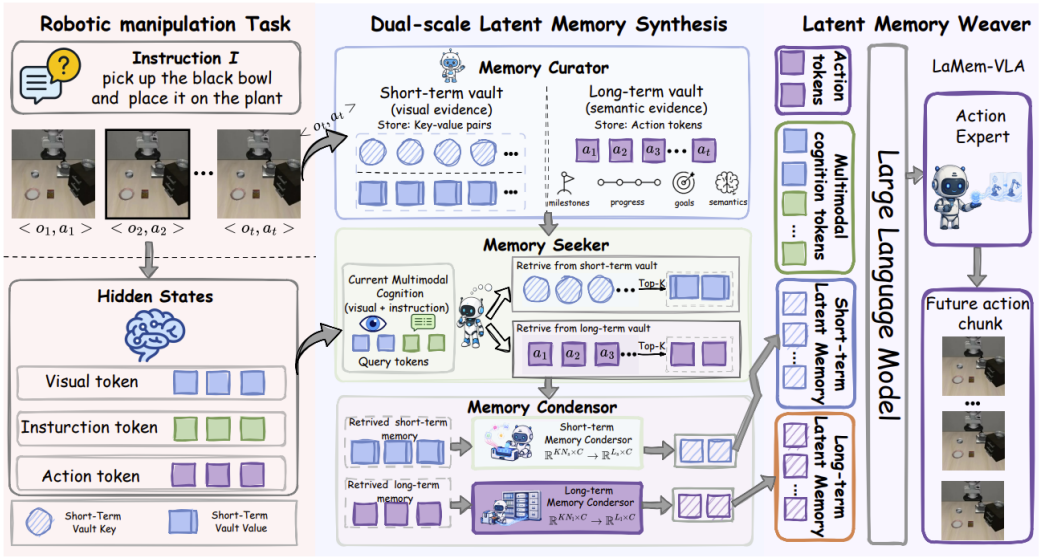

Figure 2: **The Framework of LaMem-VLA.** Given an instruction and the current observation, the vision–language encoder first encodes the inputs into a multimodal representation. The memory curator (§3.3) organizes historical experience into dual memory vaults, and the memory seeker (§3.4) then retrieves task-relevant evidence from dual memory vaults based on this multimodal representation. This retrieved evidence is compressed into fixed-length latent memory tokens by the memory condenser (§3.4). Finally, the memory weaver (§3.5) injects these latent memory tokens into the reasoning sequence, producing memory-grounded action tokens that guide the action expert to generate future action chunks.

auxiliary memory bank, and retrieves a compact set of task-relevant evidence from this bank to condition current action prediction. Since memory access is driven primarily by the current query, retrieval may become unreliable when the present observation provides weak cues or contains visually similar distractors. More fundamentally, the retrieved memory is typically consumed as auxiliary policy-side context rather than integrated into the VLA model's native token-level reasoning process, limiting its ability to directly shape action formation. In contrast, **LaMem-VLA** formulates historical experience as context-native latent memory. It maintains complementary short-term visual and long-term semantic memory vaults. Decision-relevant evidence is then retrieved and distilled into compact latent memory tokens. These latent memory tokens are directly interwoven with current visual, language, and action tokens before action formation, enabling fine-grained perceptual evidence and long-range task progress to jointly shape the VLA model's native reasoning process without incurring growing context.

3 METHOD

3.1 PROBLEM FORMULATION

We formulate the robotic manipulation task in VLA models as a language-conditioned Markovian decision-making problem. At each timestep t, the VLA policy [45, 2, 15], $\mathbf{\Pi}_\theta$ takes a natural-language instruction $\boldsymbol{I}$ and the current visual observation $\boldsymbol{o}_t$ as input, and predicts a chunk of future actions for executing the specified task:

$$\boldsymbol{a}_{t:t+H-1} = \mathbf{\Pi}_\theta(\boldsymbol{o}_t, \boldsymbol{I}), \tag{1}$$

where H denotes the action horizon, and each action $a_t \in \mathbb{R}^7$ is a 7-DoF end-effector control vector, consisting of 3-DoF relative translation, 3-DoF relative rotation, and a 1-DoF gripper state.

3.2 LAMEM-VLA MODEL ARCHITECTURE

Motivation. The Markovian formulation above in VLA models makes action prediction primarily depend on the instantaneous observation and instruction. Although effective for short-horizon behaviors, this paradigm can induce a temporal short-horizon bias in long-horizon and temporally dependent manipulation tasks, where historical state transitions, completed subtasks, and task-progress

cues are essential for reliable control. Existing works attempt to augment VLA or visuomotor policies with memory through incorporating explicit episode context into the input [56, 57, 20] or externalized memory bank retrieval [23, 24]. However, they store the historical memory outside the model's native token space and consume it as auxiliary policy-side context for action decoding, preventing memory from being fluidly interleaved with the internal reasoning process that jointly perceives, reasons, and forms action intent. Thus, our **LaMem-VLA** reformulates robotic memory as context-native latent memory to bridge this gap.

Overview. **LaMem-VLA** is an end-to-end native latent memory framework for robotic manipulation that directly weaves dual-scale historical experience into VLA reasoning to refine action generation, as shown in Fig. 2. At each timestep t, given the current visual observation $\boldsymbol{o}_t$ and instruction $\boldsymbol{I}$, the vision-language backbone embeds them into the visual tokens $\boldsymbol{X}_t$ and instruction tokens $\boldsymbol{I}$. Learnable action queries are appended to the token sequence to obtain manipulation-relevant latent action representations. **LaMem-VLA** closes the loop between latent memory reconstruction and action reasoning through four coordinated modules. First, a **latent memory curator** dynamically establishes and updates two complementary memory vaults: the short-term memory vault $\mathcal{M}^{\text{short}}$ stores visual tokens that preserve recent perceptual evidence, while the long-term memory vault $\mathcal{M}^{\text{long}}$ stores action tokens that preserve semantic and action-continuity evidence across longer horizons. Second, a **latent memory seeker** constructs a context-aware query $\boldsymbol{Q}_t^{\text{con}}$ from the current multimodal cognition representation and uses it to retrieve decision-relevant history from the two vaults. Third, a **latent memory condenser** distills the retrieved raw memory content and learnable memory tokens into compact latent short-term memory tokens $\boldsymbol{M}^{\text{short}}$ and latent long-term memory tokens $\boldsymbol{M}^{\text{long}}$, respectively. Finally, a **latent memory weaver** prepends these two sources of latent memory tokens to the current image and language tokens, forming a memory-augmented VLA input sequence $\boldsymbol{S}_t$:

$$\boldsymbol{S}_t = [\boldsymbol{M}^{\text{short}}; \boldsymbol{M}^{\text{long}}; \boldsymbol{X}_t; \boldsymbol{I}; \boldsymbol{Q}^{\text{action}}], \tag{2}$$

where $\boldsymbol{Q}^{\text{action}}$ denotes learnable action tokens. The resulting action tokens are then fed into a diffusion-based action expert [34] to generate action sequences.

3.3 Latent Memory Curator

The latent memory curator maintains the historical evidence that will later be retrieved, condensed, and woven into the VLA reasoning sequence. **LaMem-VLA** adopts a 7B-parameter Prismatic vision-language model [58] as the backbone, which is further pretrained on the large-scale cross-embodiment real robotic dataset Open-X Embodiment [11]. For the current RGB observation, the backbone first extracts visual tokens with its vision encoder [59, 60] and projects them into the language embedding space to obtain final visual tokens $\boldsymbol{X}_t \in \mathbb{R}^{N_{\text{i}} \times C}$, where N_{i} is the sequence length and C is the hidden dimension. These visual tokens are concatenated with the tokenized instruction and processed by LLaMA-7B [61]. We append learnable action queries $\boldsymbol{Q}^{\text{action}} \in \mathbb{R}^{N_{\text{a}} \times C}$ to the sequence, and use their output hidden states $\boldsymbol{H}^{\text{action}}$ as compact action representations for downstream action prediction. The curator factorizes the historical evidence into two memory vaults: a short-term memory vault $\mathcal{M}^{\text{short}}$ and a long-term memory vault $\mathcal{M}^{\text{long}}$.

Short-term Memory Vault. The short-term memory vault $\mathcal{M}^{\text{short}}$ stores visual perceptual evidence from the current episode. The resulting initialized vault is denoted as $\{\boldsymbol{m}_{\text{s}}^i\}_{i=1}^{L}$, where L specifies its initial capacity. Each short-term memory unit is represented as a key-value pair $\boldsymbol{m}_{\text{s}}^i = (\boldsymbol{k}_{\text{s}}, \boldsymbol{v}_{\text{s}})$: the key $\boldsymbol{k}_{\text{s}}$ provides a concise retrieval summary of visual evidence, while the value $\boldsymbol{v}_{\text{s}}$ stores the latent short-term memory content. Specifically, at each timestep, a learnable compression module distills the current visual tokens $\boldsymbol{X}_t$ into a compact set of short-term memory tokens, and their mean-pooled representation is used as the retrieval key:

$$\boldsymbol{v}_{\text{s}} = \mathcal{C}_{\text{s}}(\boldsymbol{X}_t) \in \mathbb{R}^{N_{\text{s}} \times C}, \quad \boldsymbol{k}_{\text{s}} = \text{MeanPool}(\boldsymbol{v}_{\text{s}}) \in \mathbb{R}^{C}, \tag{3}$$

where $\mathcal{C}_{\text{s}}$ is an SE-bottleneck compression module [62], and the unit $\boldsymbol{m}_{\text{s}}$ is then appended to $\mathcal{M}^{\text{short}}$.

Long-term Memory Vault. The long-term memory vault $\mathcal{M}^{\text{long}}$ stores the action hidden states that track task progress and action continuity across long horizons. At each timestep, the curator directly writes the action hidden states $\boldsymbol{H}^{\text{action}}$ into the vault as one long-term memory unit $\boldsymbol{m}_{\text{l}}$. Unlike the short-term vault, $\mathcal{M}^{\text{long}}$ is not a key-value bank: each long-term unit preserves the action hidden state at each timestep, allowing the vault to accumulate task-progress and action-continuity information across the trajectory.

Memory Vault Updating Strategy. After a new memory unit is written into its corresponding vault, the memory curator applies a compression strategy only when the number of stored units exceeds the capacity L. For the short-term visual stream, let $\mathcal{M}^{\text{short}} = \{\boldsymbol{m}_{\text{s}}^i = (\boldsymbol{k}_{\text{s}}^i, \boldsymbol{v}_{\text{s}}^i)\}_{i=1}^{n_{\text{s}}}$ after insertion. If $n_{\text{s}} > L$, we compute the cosine similarity between temporally adjacent keys and select the most redundant adjacent pair:

$$i_{\text{s}}^* = \arg\max_{1 \le i < n_{\text{s}}} \cos(\boldsymbol{k}_{\text{s}}^i, \boldsymbol{k}_{\text{s}}^{i+1}). \tag{4}$$

The selected pair is consolidated by averaging both its key and value tokens:

$$\tilde{\boldsymbol{k}}_{\text{s}} = \frac{1}{2}(\boldsymbol{k}_{\text{s}}^{i_{\text{s}}^*} + \boldsymbol{k}_{\text{s}}^{i_{\text{s}}^*+1}), \qquad \tilde{\boldsymbol{v}}_{\text{s}} = \frac{1}{2}(\boldsymbol{v}_{\text{s}}^{i_{\text{s}}^*} + \boldsymbol{v}_{\text{s}}^{i_{\text{s}}^*+1}), \qquad \tilde{\boldsymbol{m}}_{\text{s}} = (\tilde{\boldsymbol{k}}_{\text{s}}, \tilde{\boldsymbol{v}}_{\text{s}}). \tag{5}$$

The two adjacent units are replaced by $\tilde{\boldsymbol{m}}_{\text{s}}$, reducing redundancy while preserving their shared visual evidence. **LaMem-VLA** applies the same memory updating strategy to the long-term memory vault. See more details in Appendix.

3.4 Learning to synthesize latent memory with Dual-scale Vault

LaMem-VLA does not expose memory vaults to the action policy as raw auxiliary context. Instead, it treats the two vaults as latent evidence substrates: given the current multimodal cognition representation, **LaMem-VLA** first retrieves relevant evidence from the two vaults and then reconstructs it into compact memory tokens that are native to the VLA embedding space. This design decouples memory storage from memory consumption, allowing historical experience to remain flexible in the vaults while entering action reasoning through a bounded latent interface.

Latent Memory Seeker. The latent memory seeker retrieves evidence from the memory vaults according to the current multimodal cognition context rather than the visual observation or language instruction alone. Given the current visual tokens $\boldsymbol{X}_t$ and instruction tokens $\boldsymbol{I}$, the VLA backbone produces context-aware query $\boldsymbol{Q}_t^{\text{con}}$ that encode the current visual-linguistic cognition. The seeker then appends learnable query slots $\boldsymbol{Q}^{\text{init}} \in \mathbb{R}^{K_{\text{q}} \times C}$ to $\boldsymbol{Q}_t^{\text{con}}$ and updates only the query slots with a lightweight query builder:

$$\boldsymbol{Q}_t = \mathcal{B}([\boldsymbol{Q}_t^{\text{con}}; \boldsymbol{Q}^{\text{init}}])[-K_{\text{q}}:] \in \mathbb{R}^{K_{\text{q}} \times C}, \tag{6}$$

where $\mathcal{B}$ is a transformer-based query builder (more details in Appendix) and $\boldsymbol{Q}_t$ serves as the shared memory hook for both vaults. We apply masked attention inside $\mathcal{B}$ such that the appended query slots read from $\boldsymbol{Q}_t^{\text{con}}$, while the original multimodal hidden states are not perturbed by the query slots. The mean-pooled query $\boldsymbol{q}_t = \text{MeanPool}(\boldsymbol{Q}_t) \in \mathbb{R}^C$ is used as the global retrieval vector.

The memory seeker then uses $\boldsymbol{q}_t$ to retrieve context-relevant units from the short-term vault $\mathcal{M}^{\text{short}} = \{(\boldsymbol{k}_{\text{s}}^i, \boldsymbol{v}_{\text{s}}^i)\}_{i=1}^{|\mathcal{M}^{\text{short}}|}$ by cosine similarity:

$$\mathcal{I}_{\text{s}} = \text{Top–}K\left(\{\cos(\boldsymbol{q}_t, \boldsymbol{k}_{\text{s}}^i)\}_{i=1}^{|\mathcal{M}^{\text{short}}|}\right), \quad \boldsymbol{Z}^{\text{short}} = \text{Concat}_{i \in \mathcal{I}_{\text{s}}}(\boldsymbol{v}_{\text{s}}^i) \in \mathbb{R}^{KN_{\text{s}} \times C}, \tag{7}$$

where K is the number of retrieved short-memory units, and the discrete Top–K operation is not optimized by gradient descent. For long-term memory retrieval, the seeker similarly ranks the units in $\mathcal{M}^{\text{long}} = \{\boldsymbol{m}_{\text{l}}^i\}_{i=1}^{|\mathcal{M}^{\text{long}}|}$ via comparing the mean-pooled long-term units with the mean-pooled query $\boldsymbol{q}_t$. It then selects the Top–K long-term memory units and stacks them into $\boldsymbol{Z}^{\text{long}} \in \mathbb{R}^{KN_{\text{l}} \times C}$. The retrieved sets $\boldsymbol{Z}^{\text{short}}$ and $\boldsymbol{Z}^{\text{long}}$ are passed to the memory condenser below as short-term visual evidence and long-term semantic evidence, respectively, instead of being inserted verbatim into the VLA sequence.

Latent Memory Condenser. The retrieved short-term visual evidence $\boldsymbol{Z}^{\text{short}} \in \mathbb{R}^{KN_{\text{s}} \times C}$ and long-term semantic evidence $\boldsymbol{Z}^{\text{long}} \in \mathbb{R}^{KN_{\text{l}} \times C}$ often contain redundant historical evidence that is not fully aligned with the current context. Additionally, directly inserting these lengthy retrieved evidence sequences into the VLA sequence would expand the reasoning context and introduce redundant historical tokens, so the latent memory condenser reconstructs them into fixed-length latent memory tokens and maps them into the VLA reasoning embedding space. Specifically, we introduce learnable short-term visual memory slots $\boldsymbol{T}_{\text{s}} \in \mathbb{R}^{L_{\text{s}} \times C}$ and long-term semantic memory slots $\boldsymbol{T}_{\text{l}} \in \mathbb{R}^{L_{\text{l}} \times C}$, and update them with lightweight memory formers conditioned on the context query tokens $\boldsymbol{Q}_t \in \mathbb{R}^{K_{\text{q}} \times C}$ and the retrieved evidence:

$$\boldsymbol{M}^{\text{short}} = \mathcal{F}_{\text{v}}([\boldsymbol{Q}_t; \boldsymbol{Z}^{\text{short}}; \boldsymbol{T}_{\text{s}}])[-L_{\text{s}}:], \quad \boldsymbol{M}^{\text{long}} = \mathcal{F}_{\text{c}}([\boldsymbol{Q}_t; \boldsymbol{Z}^{\text{long}}; \boldsymbol{T}_{\text{l}}])[-L_{\text{l}}:], \tag{8}$$

where $\mathcal{F}_{\text{v}}$ and $\mathcal{F}_{\text{c}}$ denote lightweight transformer-style memory formers similar to $\mathcal{B}$ for the short-term visual and long-term semantic memory, respectively. The resulting $\boldsymbol{M}^{\text{short}}$ and $\boldsymbol{M}^{\text{long}}$ are query-conditioned latent short-term and long-term memory tokens in the same C-dimensional embedding space used by VLA reasoning. This fixed-length property makes the injected memory independent of the retrieval size while preserving evidence from the two vaults.

3.5 Interweaving Memory into Action Reasoning in Latent Space

Latent Memory Weaver for Guidance Generation. Rather than passing the condensed memory tokens to a policy head as external conditioning alone, the latent memory weaver injects the synthesized memory into the VLA reasoning sequence before action queries are resolved. Although the two types of latent memory differ in provenance, they share the same latent token interface. Given the condensed short-term memory tokens $\boldsymbol{M}^{\text{short}}$, long-term memory tokens $\boldsymbol{M}^{\text{long}}$, the weaver constructs the memory-augmented VLA input sequence $\boldsymbol{S}_t$:

$$\boldsymbol{S}_t = [\boldsymbol{M}^{\text{short}} + \mathbf{1}_{L_{\text{s}}}\boldsymbol{b}_{\text{s}}^{\top}; \boldsymbol{M}^{\text{long}} + \mathbf{1}_{L_{\text{l}}}\boldsymbol{b}_{\text{l}}^{\top}; \boldsymbol{X}_t; \boldsymbol{I}; \boldsymbol{Q}^{\text{action}}], \quad \boldsymbol{Z}^{\text{action}} = \text{VLM}(\boldsymbol{S}_t)[-N_{\text{a}}:], \tag{9}$$

where $\boldsymbol{b}_{\text{s}}, \boldsymbol{b}_{\text{l}} \in \mathbb{R}^C$ are learnable source embeddings for the two memory streams, and $\mathbf{1}_{L_{\text{s}}}, \mathbf{1}_{L_{\text{l}}}$ are all-one column vectors that broadcast them over the short-term and long-term memory tokens. Because the two sources of memory tokens are part of the model input sequence, they participate in self-attention with the current observation, language instruction, and action queries. Thus, the resulting $\boldsymbol{Z}^{\text{action}}$ inside the model reasoning process is formed as memory-grounded action tokens rather than by external policy-side fusion.

Diffusion-based Action Expert. After the vision-language backbone produces the memory-grounded action tokens $\boldsymbol{Z}^{\text{action}}$, the diffusion-based action expert decodes them into a continuous action chunk. Following the diffusion policy [8], we formulate action generation as conditional denoising over an action chunk. Let $\boldsymbol{a}^0_{t:t+H-1}$ denote the clean expert action chunk and $\boldsymbol{a}^n_{t:t+H-1}$ denote its noisy version at diffusion step n. At each denoising step, the noisy action tokens are injected with the diffusion timestep embedding and conditioned on action tokens $\boldsymbol{Z}^{\text{action}}$.

The diffusion expert ϵ_θ is trained with mean squared error (MSE) loss to predict the injected noise under these action and memory conditions:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{n,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\boldsymbol{a}^n_{t:t+H-1}, n, \boldsymbol{Z}^{\text{action}})\right\|_2^2\right]. \tag{10}$$

During inference, DDIM sampling [63] iteratively denoises the action chunk under the same conditions, producing history-aware continuous 7-DoF control actions.

4 Experiment

4.1 Implementation Details

We instantiate **LaMem-VLA** with a 7B-parameter Prismatic VLM backbone and a diffusion action expert with approximately 300M parameters. **LaMem-VLA** receives a single third-person RGB observation resized to 224×224 together with the language instruction, and predicts continuous 7-DoF end-effector actions with an action chunk size of 16. We train the model on 8 NVIDIA H800 GPUs using PyTorch FSDP. Each GPU processes 32 samples, resulting in a global batch size of 256, and the learning rate is set to 2×10^{-5}. Both the short-term and long-term memory vaults keep a maximum capacity of $L = 16$ units. The latent memory seeker retrieves the top $K = 8$ units from each vault, and the latent memory condenser reconstructs the retrieved evidence into $L_{\text{s}} = 8$ short-term memory tokens and $L_{\text{l}} = 4$ long-term memory tokens. The query builder $\mathcal{B}$ and the two memory formers $\mathcal{F}_{\text{v}}$ and $\mathcal{F}_{\text{c}}$ are implemented as two-layer transformer blocks with masked attention (more details in Appendix). During inference, the diffusion action expert generates actions with DDIM [63] sampling using 10 denoising steps.

4.2 Simulated Evaluation on SimplerEnv-Bridge

Training and Evaluation Protocol. SimplerEnv evaluates real-to-sim generalization of robot manipulation policies trained on real-world data. We evaluate **LaMem-VLA** on the SimplerEnv-Bridge [36] suite under the standard Bridge protocol. The policy is trained on the Bridge v2

Table 1: **Quantitative comparison results on SimplerEnv-Bridge [36]** (§4.2) with WidowX robot.

Method	Publication	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg. Success
RT-1-X [11]	ICRA'24	0.0	4.2	0.0	0.0	1.1
OpenVLA [2]	CoRL'25	4.2	0.0	0.0	12.5	4.2
TraceVLA [53]	ICLR'25	12.5	16.6	16.6	65.0	27.7
SpatialVLA [64]	RSS'25	16.7	25.0	29.2	100.0	42.7
Magma [65]	CVPR'25	37.5	29.2	20.8	91.7	44.8
CogACT [15]	ArXiv'24	58.3	45.8	29.2	95.8	57.3
π_0 [1]	CoRL'25	83.8	52.5	52.5	87.9	69.2
DreamVLA [66]	NeurIPS'25	45.8	45.8	25.0	87.5	51.0
ThinkAct [67]	NeurIPS'25	58.3	37.5	8.7	70.8	43.8
CronusVLA [68]	AAAI26	66.7	54.2	20.8	100.0	60.4
MemoryVLA [23]	ICLR'26	75.0	75.0	37.5	100.0	71.9
SemanticVLA [69]	CVPR'26	83.6	54.5	40.3	81.8	65.1
LaMem-VLA (Ours)	–	83.3	75.0	41.7	95.8	**73.9**

dataset [13] for 50k optimization steps, with validation performed every 2.5k steps. The final results are reported using the checkpoint that achieves the best validation performance. For evaluation, each task is tested over 24 trials, and we report the average success rate across trials.

Evaluation Results. As shown in Table 1, **LaMem-VLA** achieves the highest average success rate of **73.9**% on SimplerEnv-Bridge, yielding a **16.6** gain over the CogACT [15] baseline and surpassing recent state-of-the-art VLAs such as π_0 [1] and SemanticVLA [69]. Per task, **LaMem-VLA** obtains **83.3**% *on Put Spoon on Towel*, **75.0**% on *Put Carrot on Plate*, **41.7**% on *Stack Cube*, and **95.8**% on *Put Eggplant in Basket*. These results indicate that injecting dual latent memory tokens into the VLA reasoning sequence provides effective historical context for action prediction, improving manipulation performance and maintaining strong robustness across diverse task settings.

4.3 Simulated Evaluation on LIBERO

Training and Evaluation Protocol. We evaluate **LaMem-VLA** on the LIBERO [35] benchmark using a Franka robot across five suites: spatial awareness (Spatial), object manipulation (Object), goal completion (Goal), and long-horizon reasoning (Long-10 and Long-90). The first four suites comprise 10 tasks each, while Long-90 contains 90 tasks. Following the OpenVLA protocol [2], we use 50 demonstrations per task. We train separate models for Spatial, Object, and Goal for 20k optimization steps, and jointly train on Long-10 and Long-90 for 40k steps. During training, we evaluate checkpoints at 1k-step intervals and select the best-performing one according to validation success. We report success rates for each suite and the overall average over 50 rollouts per task.

Evaluation Results. As shown in Table 2, **LaMem-VLA** achieves the best overall performance on LIBERO, reaching an average success rate of **97.6**% across the five suites. It improves over the strongest reported memory-augmented method MemoryVLA [23], by **1.1** points, and surpasses strong VLA baselines such as CogACT [15] by **4.4** points. Compared with π_0 [1], **LaMem-VLA** obtains a first-four-suite average of **97.7**%, yielding a **3.5** point improvement. Notably, these gains are achieved without the additional proprioceptive and wrist-camera inputs used by starred methods. Across individual suites, **LaMem-VLA** consistently attains the highest success rates, with **98.8**% on Spatial, **99.0**% on Object, **97.2**% on Goal, **95.8**% on Long-10, and **97.0**% on Long-90. The improvements on the long-horizon suites are particularly important: **LaMem-VLA** outperforms MemoryVLA by **2.4** points on Long-10 and **1.4** points on Long-90. These results suggest that weaving short-term visual evidence and long-term semantic memory directly into the VLA reasoning sequence helps the policy resolve temporally ambiguous states, preserve task progress, and generate more reliable actions for both general manipulation and long-horizon instruction following.

4.4 Ablation Study

Effectiveness of Dual-scale Latent Memory. We evaluate the contribution of each memory source by selectively removing it from the latent VLA input sequence. We consider four settings: the full **LaMem-VLA** with both short-term and long-term latent memory, *w/o Short-term Memory* that

Table 2: **Quantitative comparison results on LIBERO [35] (§4.3) with Franka robot.** Success rates (%) are reported across five suites. * indicates methods using additional proprioceptive and wrist-camera inputs. For methods without LIBERO-90 results, we report the average over the first four suites.

Method	Publication	Spatial	Object	Goal	Long-10	Long-90	Avg. Success
Diffusion Policy [8]	RSS'23	78.3	92.5	68.3	50.5	–	72.4
Octo [10]	RSS'24	78.9	85.7	84.6	51.1	–	75.1
UniACT [70]	CVPR'25	77.0	87.0	77.0	70.0	73.0	76.8
SpatialVLA [64]	RSS'25	88.2	89.9	78.6	55.5	46.2	71.7
OpenVLA [2]	CoRL'25	84.7	88.4	79.2	53.7	73.5	75.9
CoT-VLA [40]	CVPR'25	87.5	91.6	87.6	69.0	–	83.9
π_0-FAST* [71]	ArXiv'25	96.4	96.8	88.6	60.2	83.1	85.0
CogACT [15]	ArXiv'24	97.2	98.0	90.2	88.8	92.1	93.2
π_0^* [1]	CoRL'25	96.8	98.8	95.8	85.2	–	94.2
ThinkAct [67]	NeurIPS'25	88.3	91.4	87.1	70.9	–	84.4
MemoryVLA [23]	ICLR'26	98.4	98.4	96.4	93.4	95.6	96.5
Fast-ThinkAct [72]	CVPR'26	92.0	97.2	90.2	79.4	–	89.7
SemanticVLA [69]	CVPR'26	98.0	98.6	96.8	94.4	–	97.0
LARA [73]	ICML'26	96.5	97.5	96.0	92.5	–	95.6
LaMem-VLA (Ours)	–	98.8	99.0	97.2	95.8	97.0	**97.6**

Table 3: **Detailed analysis of dual-scale latent memory** on SimplerEnv [36] and LIBERO [35] (§4.4).

Method	SimplerEnv	LIBERO-90
w/o Dual Memory	57.3	92.1
w/o Short-term Memory	65.6	95.4
w/o Long-term Memory	64.6	94.8
LaMem-VLA (Ours)	**73.9**	**97.0**

Table 4: **Ablation study of latent-native memory integration** on SimplerEnv [36] and LIBERO [35] (§4.4).

Method	SimplerEnv	LIBERO-90
BASELINE	57.3	92.1
Policy-side Memory	71.9	94.8
Raw Retrieval Conditioning	69.8	95.1
LaMem-VLA (Ours)	**73.9**	**97.0**

removes the visual-dominant memory tokens and only keeps long-term semantic memory, *w/o Long-term Memory* that removes the long-term semantic memory tokens and only keeps short-term visual memory, and *w/o Dual-scale Memory* that removes both memory streams. Table 3 shows that the full dual-memory design achieves the best performance on both benchmarks, reaching **73.9**% on SimplerEnv [36] and **97.0**% on LIBERO-90 [35]. Removing both memory streams causes the largest degradation, dropping performance to 57.3% and 92.1%, respectively. These results suggest the complementary roles of the two vaults: short-term memory preserves current-episode visual evidence such as object states and transient perceptual cues, while long-term memory maintains task progress and action-continuity evidence over longer horizons. Therefore, removing either stream should weaken temporally grounded action reasoning, and removing both streams should further reduce **LaMem-VLA** to a memory-free VLA policy.

Latent-native Memory *vs.* Policy-side Conditioning. We isolate the effect of memory consumption path in Table 4, where BASELINE denotes the memory-free action policy without short-term and long-term memory. The baseline reaches only 57.3% on SimplerEnv [36] and 92.1% on LIBERO-90 [35], while adding memory as an external policy-side condition improves performance to 71.9% and 94.8%, respectively. Directly feeding raw retrieved evidence to the action policy also improves over the baseline, but remains below the full model **LaMem-VLA**, suggesting that uncompressed retrieval can introduce redundant historical tokens. By prepending compressed memory tokens as a memory-augmented VLA input sequence, **LaMem-VLA** enables action tokens to attend to latent memory, observation, and instruction in the same embedding space, achieving the best results of **73.9**% and **97.0**%. These results confirm that the gain comes from latent-native memory integration rather than merely adding memory as external context.

The Number of Retrieved Memory Units K. We further study how the retrieval budget of the memory seeker affects manipulation performance by varying the number of retrieved memory units per vault (Eq. 7). Specifically, we evaluate $K \in \{2, 4, 8, 12\}$ to examine how much historical evidence should be exposed to the condenser. As shown in Table 5, increasing the retrieval budget from $K = 2$ to $K = 4$ improves SR from 66.7% to 70.8% on SimplerEnv [36] and from 94.4% to 95.9% on LIBERO-90 [35]. Further

Table 5: **Ablation study of the retrieved memory unit number K** on SimplerEnv [36] and LIBERO-90 [35] (§4.4).

K	SimplerEnv	LIBERO-90
2	66.7	94.4
4	70.8	95.9
8	**73.9**	**97.0**
12	71.8	96.2

increasing the budget to $K = 8$ yields the best performance, reaching **73.9**% on SimplerEnv and **97.0**% on LIBERO-90. This indicates that a very small retrieval budget provides insufficient historical evidence for recovering task progress and transient visual changes. However, using $K = 12$ slightly reduces performance to 71.8% and 96.2%. These results suggest that retrieving more memory units is beneficial to a moderate budget, while excessive retrieval can introduce redundant or weakly related evidence and increase the compression burden of the memory condenser.

The Number of Latent Memory Tokens L_s and L_l. We next investigate the impact of the short-term and long-term latent memory token number (Eq. 8) in Fig. 3. For short-term memory, increasing L_s from 2 to 16 raises SimplerEnv [36] SR from 61.4% to 65.6%, suggesting that more short-term latent memory tokens help preserve fine-grained perceptual evidence; the gain saturates when $L_s = 32$. For long-term memory, fixing $L_s = 8$ and increasing long-term latent memory token number L_l provides stronger task-progress and action-continuity cues, reaching up to 73.9% on SimplerEnv [36] and 97.0% on LIBERO-90 [35]. Although larger latent memory token budgets can yield strong performance, they also lengthen the VLA self-attention context and increase computational costs; therefore, we use $(L_s, L_l) = (8, 4)$ as a balanced default between performance and efficiency.

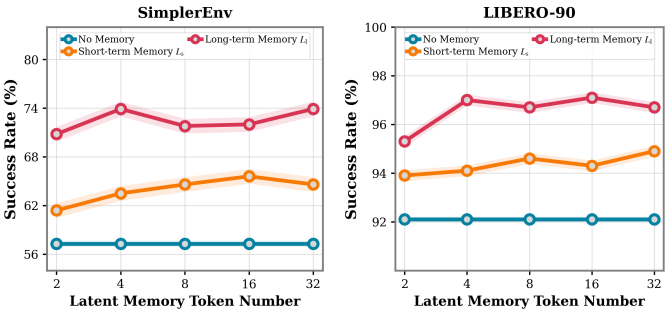

Figure 3: **Ablation study of the latent memory token number** on SimplerEnv [36] and LIBERO-90 [35] (§4.4).

5 Conclusion

In this paper, we propose **LaMem-VLA**, a dual latent memory framework for reducing the temporal short-horizon bias of vision-language-action models. The core idea is to make robotic history part of the model's native latent context, rather than an external condition attached after multimodal reasoning. To this end, **LaMem-VLA** uses complementary short-term visual memory and long-term semantic memory, and represents the retrieved history as compact latent memory tokens inside the VLA input sequence. This allows past visual evidence and task-progress cues to interact with the current observation, language instruction, and action queries through the same embedding space. Extensive experiments in two simulation platforms demonstrate the superior performance of **LaMem-VLA**. These results suggest that context-native latent memory is a practical direction for building VLA systems with stronger temporal awareness.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 2, 3, 8, 9

[2] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *CoRL*, pages 2679–2713, 2025. 1, 2, 3, 4, 8, 9

[3] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *ICLR*, volume 2025, pages 29982–30009, 2025. 1, 3

[4] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. In *ICML*, pages 61229–61245, 2024. 1

[5] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 1

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1

[7] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. On scaling up a multilingual vision and language model. In *CVPR*, pages 14432–14444, 2024. 1

[8] 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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. 1, 2, 7, 9

[9] Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *AAAI*, volume 39, pages 14754–14762, 2025. 1, 3

[10] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *RSS*, 2024. 1, 9

[11] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *ICRA*, pages 6892–6903, 2024. 1, 5, 8

[12] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics*. 1

[13] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, pages 1723–1736, 2023. 1, 8

[14] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 1

[15] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024. 1, 2, 3, 4, 8, 9

[16] Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, Zhaopeng Chen, and Alois Knoll. Language-guided object-centric diffusion policy for generalizable and collision-aware robotic manipulation. *arXiv preprint arXiv:2407.00451*, 2024. 2

[17] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *RSS*, 2024. 2

[18] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 2, 3

[19] Cunxin Fan, Xiaosong Jia, Yihang Sun, Yixiao Wang, Jianglan Wei, Ziyang Gong, Xiangyu Zhao, Masayoshi Tomizuka, Xue Yang, Junchi Yan, et al. Interleave-vla: Enhancing robot manipulation with interleaved image-text instructions. In *CoRL*, 2025. 2, 3

[20] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, et al. Cronusvla: Towards efficient and robust manipulation via multi-frame vision-language-action modeling. *arXiv preprint arXiv:2506.19816*, 2025. 2, 3, 5

[21] Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Hamlet: Switch your vision-language-action model into a history-aware policy. *arXiv preprint arXiv:2510.00695*, 2025. 2, 3

[22] Qingda Hu, Ziheng Qiu, Zijun Xu, Kaizhao Zhang, Xizhou Bu, Zuolei Sun, Bo Zhang, Jieru Zhao, Zhongxue Gan, and Wenchao Ding. Resolving state ambiguity in robot manipulation via adaptive working memory recoding. *IEEE Robotics and Automation Letters*, 2026. 2, 3

[23] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *ICLR*, 2026. 2, 3, 5, 8, 9

[24] Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Memer: Scaling up memory for robot control via experience retrieval. *arXiv preprint arXiv:2510.20328*, 2025. 2, 3, 5

[25] Zaijing Li, Bing Hu, Rui Shao, Gongwei Chen, Dongmei Jiang, Pengwei Xie, Jianye Hao, and Liqiang Nie. Global prior meets local consistency: Dual-memory augmented vision-language-action model for efficient robotic manipulation. In *CVPR*, pages 35135–35145, 2026. 2, 3

[26] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *CVPR*, pages 12030–12040, 2026. 2

[27] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *CVPR*, pages 33510–33520, 2026. 2

[28] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *ICLR*, 2026. 2

[29] Shuanghao Bai, Jing Lyu, Wanqi Zhou, Zhe Li, Dakai Wang, Lei Xing, Xiaoguang Zhao, Pengwei Wang, Zhongyuan Wang, Cheng Chi, et al. Latent reasoning vla: Latent thinking and prediction for vision-language-action models. *arXiv preprint arXiv:2602.01166*, 2026. 2

[30] Guibin Zhang, Muxin Fu, and Shuicheng Yan. Memgen: Weaving generative latent memory for self-evolving agents. *arXiv preprint arXiv:2509.24704*, 2025. 2

[31] Xinlei Yu, Chengming Xu, Guibin Zhang, Zhangquan Chen, Yudong Zhang, Yongbo He, Peng-Tao Jiang, Jiangning Zhang, Xiaobin Hu, and Shuicheng Yan. Vismem: Latent vision memory unlocks potential of vision-language models. In *CVPR*, pages 31544–31555, 2026. 2

[32] Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Guanting Dong, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026. 2

[33] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pages 2165–2183, 2023. 2, 3

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 2, 5

[35] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, volume 36, pages 44776–44791, 2023. 2, 8, 9, 10

[36] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *CoRL*, pages 3705–3728. PMLR, 2025. 2, 7, 8, 9, 10

[37] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. In *CoRL*, pages 5326–5350, 2025. 3

[38] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *CoRL*, pages 4573–4602, 2025. 3

[39] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025. 3

[40] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *CVPR*, pages 1702–1713, 2025. 3, 9

[41] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. 3

[42] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025. 3

[43] Juyi Lin, Amir Taherin, Arash Akbari, Arman Akbari, Lei Lu, Guangyu Chen, Taskin Padir, Xiaomeng Yang, Weiwei Chen, Yiqian Li, et al. Vote: vision-language-action optimization with trajectory ensemble voting. *arXiv preprint arXiv:2507.05116*, 2025. 3

[44] Songsheng Wang, Rucheng Yu, Zhihang Yuan, Chao Yu, Feng Gao, Yu Wang, and Derek F Wong. Spec-vla: speculative decoding for vision-language-action models with relaxed acceptance. In *EMNLP*, pages 26916–26928, 2025. 3

[45] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. 3, 4

[46] Rongyu Zhang, Menghang Dong, Yuan Zhang, Liang Heng, Xiaowei Chi, Gaole Dai, Li Du, Dan Wang, Yuan Du, and Shanghang Zhang. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. In *AAAI*, volume 40, pages 18764–18772, 2026. 3

[47] Hongyu Wang, Chuyan Xiong, Ruiping Wang, and Xilin Chen. Bitvla: 1-bit vision-language-action models for robotics manipulation. *arXiv preprint arXiv:2506.07530*, 2025. 3

[48] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. In *CoRL*, pages 3094–3114, 2025. 3

[49] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *ICCV*, pages 11089–11099, 2025. 3

[50] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Wenhao Zhang, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025. 3

[51] Hang Li, Fengyi Shen, Dong Chen, Liudi Yang, Xudong Wang, Jinkui Shi, Zhenshan Bing, Ziyuan Liu, and Alois Knoll. Remem-vla: Empowering vision-language-action model with memory via dual-level recurrent queries. *arXiv preprint arXiv:2603.12942*, 2026. 3

[52] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *ICLR*, volume 2024, pages 26703–26721, 2024. 3

[53] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *ICLR*, volume 2025, pages 54277–54296, 2025. 3, 8

[54] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 3

[55] Max Sobol Mark, Jacky Liang, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Dhruv Shah, and Aviral Kumar. Bpp: Long-context robot imitation learning by focusing on key history frames. *arXiv preprint arXiv:2602.15010*, 2026. 3

[56] Minghui Lin, Pengxiang Ding, Shu Wang, Zifeng Zhuang, Yang Liu, Xinyang Tong, Wenxuan Song, Shangke Lyu, Siteng Huang, and Donglin Wang. Hif-vla: Hindsight, insight and foresight through motion representation for vision-language-action models. In *CVPR*, pages 20732–20742, 2026. 5

[57] Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hojin Jeon, Younggyo Seo, and Jinwoo Shin. Contextvla: Vision-language-action model with amortized multi-frame context. *arXiv preprint arXiv:2510.04246*, 2025. 5

[58] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *ICML*, 2024. 5

[59] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. 5

[60] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 5

[61] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5

[62] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 5

[63] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 7

[64] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 8, 9

[65] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *CVPR*, pages 14203–14214, 2025. 8

[66] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. In *NeurIPS*, pages 24195–24228, 2026. 8

[67] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. In *NeurIPS*, pages 82782–82802, 2026. 8, 9

[68] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, et al. Towards efficient and robust manipulation via multi-frame vision-language-action modeling. In *AAAI*, volume 40, pages 18388–18396, 2026. 8

[69] Fei Ni, Zhuo Chen, Yifu Yuan, Zibin Dong, Xianze Yao, Shan Luo, Jianye Hao, Jiankang Deng, and Stefanos Zafeiriou. Semanticvla: Towards semantic reasoning over action memorization via synergistic explicit trace and latent action planning. In *CVPR*, pages 12237–12247, 2026. 8, 9

[70] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *CVPR*, pages 22508–22519, 2025. 9

[71] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 9

[72] Chi-Pin Huang, Yunze Man, Zhiding Yu, Min-Hung Chen, Jan Kautz, Yu-Chiang Frank Wang, and Fu-En Yang. Fast-thinkact: Efficient vision-language-action reasoning via verbalizable latent planning. *arXiv e-prints*, pages arXiv–2601, 2026. 9

[73] Mengya Liu, Baoxiong Jia, Jiangyong Huang, Jingze Zhang, and Siyuan Huang. Lara: Latent action representation alignment for vision-language-action models. *arXiv preprint arXiv:2606.07100*, 2026. 9