Zeva: In-Context Causal Learning for Generalizable Embodied Manipulation

Fu Chen*, Xin Ding*,†, Bingjia Huang, Xiangyu Li, Mingju Wang, Jiawei He
Kun Li, Wei Sun, Yunxin Liu, Hao Wu†,‡, Ting Cao†,§
Institute for AI Industry Research (AIR), Tsinghua University; Z-Trans AI
*Co-first authors †Corresponding authors § Project lead ‡Work done during a visit to AIR, Tsinghua
Project Page: https://air-embodied-brain.github.io/Zeva

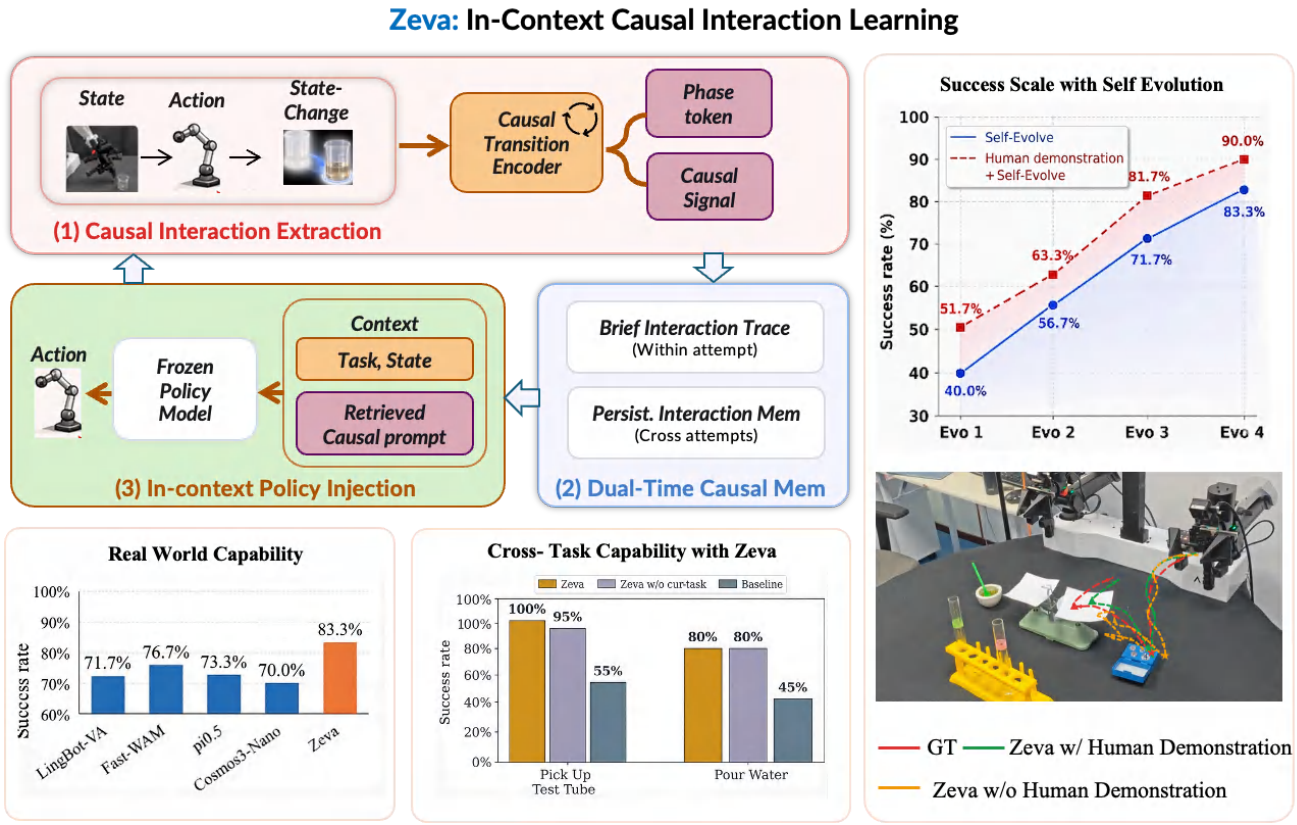


Fig. 1. Zeva enables self-evolving embodied intelligence through In-Context Causal Learning (ICCL). Zeva learns the causal relationships between robot actions and resulting state changes by extracting action-induced causal transitions, maintaining dual-timescale interaction memory, and retrieving causal context for in-context policy adaptation. Without parameter updates, Zeva enables frozen policies to progressively improve through self-evolution and human-guided experience injection, achieving strong real-world performance and cross-task knowledge transfer beyond task-specific demonstrations.

***Abstract*—Generalizable embodied manipulation remains difficult to achieve through pretraining alone, due to unseen physical conditions in the real world. We argue that robots need to learn from their own physical interactions on the fly during real-world deployment and use this knowledge to inform subsequent actions. We present Zeva, the first framework that enables in-context learning from a robot's own physical interaction experience while keeping the policy model frozen. Zeva employs a Causal Interaction Extractor to encode an executed action and its induced state change into a causal interaction signal, which is stored in a dual-timescale causal memory. For subsequent actions, relevant causal interaction signals are retrieved from memory and injected into the frozen policy model as context. Experiments in simulation and real-world manipulation demonstrate that Zeva achieves the best performance among the compared frontier VLAs and WAMs and, more importantly, enables self-evolution during deployment without gradient updates. Its success rate continues to improve as the robot accumulates interaction experience. Furthermore, the acquired interaction experience can generalize across tasks.**

***Index Terms*—Embodied Foundation Models, Interaction Memory, Test-time Scaling, Causal Representation**

I. Introduction

Generalizable interaction-intensive manipulation is the central challenge for embodied AI. Embodied foundation models,

including VLAs and WAMs, have made rapid progress towards this goal [1]–[9], yet deployment in the physical world remains brittle. At deployment, robots inevitably encounter physical conditions that are absent during pretraining, such as novel object geometries, contact dynamics, visual variations, and embodiment-specific execution errors. These conditions interact with continuous, high-dimensional robot states and actions, causing similar actions to produce significant different outcomes. It is therefore difficult for pretraining alone to cover the diversity of physical interactions.

To realize generalizable embodied manipulation, we argue that *the robot should learn from its own physical interaction experiences on the fly during execution, i.e., learning the causality between its action and the induced state change, and can apply it to subsequent actions*. Recent post-pretraining adaptation methods have begun to move in this direction, but still leave gaps. Test-time training methods, such as RoboTTT [10] and WAM-TTT [11], convert deployment histories into adaptive fast weights or memory modules through test-time gradient updates. This leaves the action and state-change structure of each interaction implicit. Recent in-context robot learners, such as GEN-1.5 [12] and Skild S1 [13], instead keep model weights fixed and infer a new task from one or a few demonstrations in context. However, this form of in-context learning primarily targets task generalization rather than on-the-fly interaction learning.

We present *Zeva*, the first framework that enables in-context learning from the robot's own physical interaction experiences, toward generalized embodied manipulation in real deployment. Zeva keeps the policy model frozen at deployment, but extracts the causality between actions and state changes during each action step and retrieves this knowledge as context for subsequent action generation. This enables the same frozen policy to adapt to diverse real-world physical conditions across repeated self-attempts and exhibit self-evolution, i.e., the more it attempts, the higher its success rate becomes.

Zeva realizes this idea as **In-Context Causal Learning** (ICCL) through three stages. First, **Causal Interaction Extraction** uses a Causal Transition Encoder (CTE) to integrate visual latents, action encodings, and observed effects into a Causal Interaction State, which is then projected into a task **Phase Token** and a **Causal Interaction Signal**. Second, **Dual-timescale Causal Memory** organizes these signals for deployment-time adaptation: a **Brief Interaction Trace** (BIT) captures recent within-attempt dynamics, while a **Persistent Interaction Memory** (PIM) consolidates useful evidence across attempts within the same episode through similarity-based merging. Third, **In-Context Policy Injection** retrieves phase-matched interaction evidence and constructs a **Causal Prompt** for the frozen foundation policy. During deployment, all model parameters remain frozen; only the interaction memories are updated online.

We evaluate Zeva on RoboCasa365-Atomic5 and a real-world chemical manipulation benchmark, *ChemLab-Evo*, using an ARX manipulator. The main results are: (1) Zeva achieves the best success rate among the compared frontier VLA and WAM models on RoboCasa365-Atomic5, reaching 76.8%. (2) Zeva achieves the best success rate on the real-world ChemLab-Evo across all three difficulty levels. (3) Zeva exhibits *success-rate scaling from its own interaction experience*: on RoboCasa365-Atomic5, the cumulative success rate increases from 26% at the first attempt to 73% within four repeated attempts, and ChemLab-Evo shows the same scaling trend across attempts. (4) Zeva also exhibits in-context learning from human teleoperation demonstrations, further improving performance by up-to 15% beyond learning from its own interactions alone.

Our contributions are summarized as follows:

- We formulate embodied action generalization in real-world deployment as In-Context Causal Learning, where a frozen robot policy learns from its own physical interaction experiences without test-time weight updates.
- We propose Zeva, which combines Causal Interaction Extraction, Dual-timescale Causal Memory, and In-Context Policy Injection for gradient-free self-evolution across repeated attempts.
- We validate Zeva in simulation and real-world manipulation, showing improved success across robot attempts and the strongest success rate compared to frontier embodied policy models.

II. Related Work

A. Vision-Language-Action and World-Action Foundation Models

VLA models combine a vision-language backbone with an action-generation head, learning an observation-to-action mapping directly from demonstration data; representative work includes OpenVLA [1], π_0 [2], and its successors $\pi_{0.5}$ [14] and $\pi^*_{0.6}$ [15]. World-Action Models (WAMs) further unify action generation with future visual prediction, absorbing physical priors through large-scale video pretraining [16], [17], adapting pretrained video diffusion models to jointly denoise future frames and actions [7], [9], or autoregressively interleaving video and action tokens under a causal attention mask [8]. All of these models require large-scale pretraining to obtain generalizable priors, and their weights remain fixed after deployment, so their capability ceiling is set by the pretraining corpus. This work adopts Cosmos3's [18] rectified-flow action-generation head as the policy backbone, but obtains continual adaptation after deployment through an external self-evolving mechanism rather than through pretraining scale.

B. Behavioral and Latent-Action Representation Learning

To mitigate distribution shift caused by high-dimensional observation-to-action mapping, one line of work [19]–[24], learns low-dimensional behavioral representations: ALAM [25] and LAPA [26] learn latent-action encodings from unlabeled video via algebraic-consistency constraints and inverse-dynamics reconstruction, respectively; BehaviorVLA [27] encodes a trajectory into a scene-agnostic global prototype and an online-updated local phase; UniVLA [28]

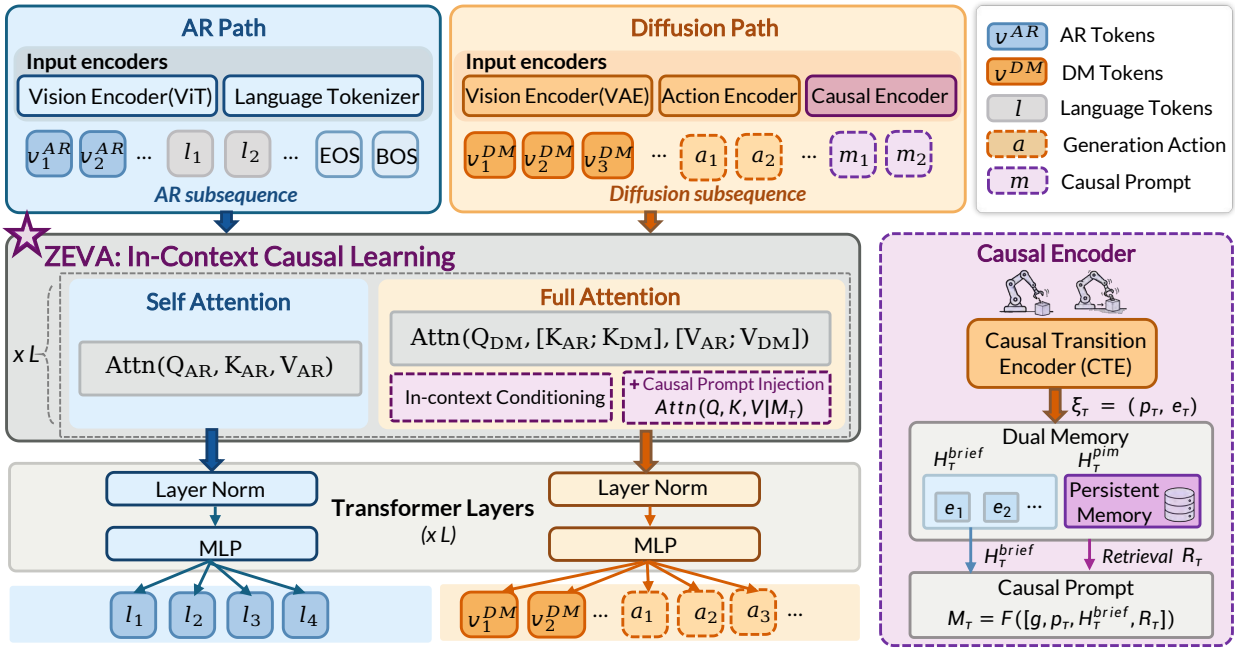


Fig. 2. Detailed architecture of the Zeva framework. Zeva integrates causal interaction intelligence into a frozen foundation policy through three main components: (1) Causal Transition Encoder, which maps action-effect pairs into latent causal signals; (2) Dual-timescale Causal Memory, consisting of a Brief Interaction Trace (BIT) for within-attempt dynamics and a Persistent Interaction Memory (PIM) for cross-attempt experience; and (3) Causal Prompt Injection, where retrieved phase-matched evidence is injected into the Transformer's self-attention layers as an in-context prompt to guide the Diffusion-based action generation.

derives task-centric latent action representations from action-label-free video across arbitrary embodiments and viewpoints. These methods show that low-dimensional behavioral representations outperform high-dimensional direct mapping, but in every case the representation space is fixed before deployment and cannot accumulate interaction experience from real deployment, making them difficult to keep effective as the distribution keeps changing.

C. Memory-Augmented Robot Policies

Another line of work equips policies with online memory to handle long-horizon dependencies. MemoryVLA [29] maintains a perceptual-cognitive memory bank and handles non-Markovian control through a retrieve-fuse-consolidate mechanism; MEM [30] combines short-term video memory with a long-term language-based event summary to support tasks lasting more than ten minutes; DIM-WAM [31] and MemoryWAM [32] compress visual history through, respectively, a multi-type historical event bank and a sliding-window-plus-anchor-frame mechanism. In each case, the memory content consists of observation- or semantic-level historical fragments. Determining which direction an action shifted the state on a failed attempt, and how it should be adjusted on the next attempt of the same episode, requires explicitly preserving the action–state-change relationship across attempt boundaries, something perception-side history alone does not capture.

D. Deployment-Time Adaptation and Experience-Driven Self-Improvement

The line of work closest to our motivation enables policies to improve from their own deployment experience. $\pi^*_{0.6}$ (RECAP) [15] and VLA-RL [33] update policies through offline or online reinforcement learning on autonomous rollouts. However, their gradient-based optimization cycles are too slow to adapt the policy between consecutive attempts and are vulnerable to capability collapse. RoboTTT [10] and WAM-TTT [11] accelerate adaptation by updating lightweight fast weights or memory modules via test-time gradients. Nevertheless, the adaptation latency remains far beyond the timescale of individual attempts, and WAM-TTT additionally relies on human demonstration videos as the update source, preventing the agent from autonomously exploring and improving through its own interaction experiences.

III. Method

A. Problem Formulation: In-Context Causal Learning

We consider an embodied agent [34]–[36] that must generalize to novel tasks $\mathcal{T} \sim P(\mathcal{T})$ by learning from interaction in-context. Unlike standard imitation learning, our goal is **In-Context Causal Learning (ICCL)**, where the agent should infer the causal structure of the environment, specifically the relationship between its actions and the resulting state changes, within a single episode without weight updates ($\nabla_\theta \pi = 0$).

Formally, at attempt r, the policy π leverages a causal context $\mathcal{M}_{r-1}$ extracted from previous interactions. The objective is to maximize the success rate on unseen tasks by

minimizing the discrepancy between the predicted and actual causal effects:

$$\mathcal{L}_{ICCL} = \mathbb{E}_\tau[\|\Delta s_\tau - \text{Inference}(\pi(o_\tau, \mathcal{M}_{\tau-1}))\|] \quad (1)$$

where $\mathcal{M}$ acts as the "causal prompt" that guides the frozen foundation model to adapt to new physical dynamics.

1) Causal Context as Interaction Evidence: To achieve this, we define the interaction experience as a sequence of **interaction triplets**. At each timestep τ, the agent action u_τ is treated as an interaction with the environment, and the resulting visual state change $\Delta s_\tau = s_{\tau+k} - s_\tau$ is its effect. A causal interaction unit ξ_τ is defined as:

$$\xi_\tau = \phi(s_\tau, u_\tau, \Delta s_\tau) \quad (2)$$

The causal context $\mathcal{M}$ is an organized collection of these units ξ. By conditioning on $\mathcal{M}$, the frozen policy $\pi(a_\tau \mid o_\tau, g, \mathcal{M})$ performs implicit system identification. It infers latent environmental properties such as object mass or joint constraints from past interaction signals, thereby generalizing its action strategy to previously unseen physical configurations.

B. Overview of Zeva

To realize the ICCL objective, we propose **Zeva**, a framework that transforms raw interactions into a dual timescale causal memory. Zeva consists of three parts: (1) **Causal Interaction Extraction**, which maps actions and effects into a latent space, (2) **Dual timescale Memory Management**, which structures experiences for retrieval, and (3) **In-Context Policy Injection**, which provides the causal prompt to the foundation model.

Figure **??** shows how these components couple to the two-path foundation policy. The autoregressive path encodes multimodal context, while the diffusion path generates continuous robot actions. For every executed transition, CTE extracts a phase token and a causal interaction signal: the Brief Interaction Trace supplies recent within-attempt evidence, whereas the Persistent Interaction Memory retrieves phase-matched evidence accumulated across attempts. The resulting causal prompt conditions the full-attention blocks of the diffusion path, allowing the policy to change its action strategy through context while keeping all model parameters fixed.

C. Part 1: Causal Interaction Extraction

To bridge the gap between raw pixels and causal reasoning, we use a **Causal Transition Encoder**(CTE) to map interactions into a latent causal space.

a) Interaction Recurrence.: At each timestep τ, we extract three transient signals: visual latents s_τ, action encoding u_τ, and the observed effect d_τ. Inspired by trajectory-level behavioral representations [27], we integrate these signals into a **Causal Interaction State** B_τ via a gated recurrence::

$$B_\tau = \text{GRU}(B_{\tau-1}, [s_\tau; u_\tau; d_\tau]), \quad B_0 = F_{\text{init}}(g, s_0) \quad (3)$$

The state B_τ is projected into two components: a **Phase Token** p_τ representing task progress and a **Causal Interaction Signal** e_τ characterizing the dynamics. Together, they form the causal interaction unit $\xi_\tau = (p_\tau, e_\tau)$ as defined in Section III-A.

b) Learning Objectives.: We optimize the Causal Transition Encoder with a multi objective loss to ensure e_τ captures physical causality:

- **Causal Effect Prediction ($\mathcal{L}_{\text{effect}}$)**: This objective decodes e_τ and candidate actions to predict visual state changes Δs_τ, grounding the signal in physical effects.
- **Task Identity Clustering ($\mathcal{L}_{\text{task}}$)**: A supervised contrastive loss that clusters episodes of the same task to learn a task consistent interaction space.
- **Phase Progression ($\mathcal{L}_{\text{phase}}$)**: A loss that ensures p_τ captures the monotonic progression of task execution.

D. Part 2: Dual timescale Causal Memory

Zeva organizes interaction units (p_τ, e_τ) into two memory streams to support adaptation both within and across attempts of the current episode.

a) Brief Interaction Trace.: $\mathcal{H}_\tau^{\text{brief}}$ stores a sliding window of the most recent interaction signals $\{e_j\}_{j=\tau-K}^{\tau}$. This provides the policy with immediate local context regarding the efficacy of current actions.

$$\mathcal{H}_\tau^{\text{brief}} = \{e_j\}_{j=\tau-K_{\text{brief}}}^{\tau} \quad (4)$$

This trace is reset at the end of each attempt.

b) Persistent Interaction Memory.: $\mathcal{H}_\tau^{\text{pim}}$ accumulates causal units $\xi = (p, e)$ across attempts of the same episode and is cleared before a new episode begins. To ensure scalability, we employ similarity based merging. For a new unit $\xi_\tau = (p_\tau, e_\tau)$, we find the most similar entry $\xi^* \in \mathcal{H}_\tau^{\text{pim}}$ by calculating:

$$S_{\text{merge}}(\xi_\tau, \xi_j) = \beta_p \text{sim}(p_\tau, p_j) + \beta_e \text{sim}(e_\tau, e_j) \quad (5)$$

The persistent memory is then updated as follows:

$$\mathcal{H}_\tau^{\text{pim}} \leftarrow \begin{cases} (\mathcal{H}_{\tau-1}^{\text{pim}} \setminus \{\xi^*\}) & \\ \quad \cup \{\text{Merge}(\xi^*, \xi_\tau)\} & \text{if } S_{\text{merge}}(\xi_\tau, \xi^*) > \tau_{\text{merge}}, \\ \mathcal{H}_{\tau-1}^{\text{pim}} \cup \{\xi_\tau\}, & \text{otherwise} \end{cases} \quad (6)$$

This ensures that a new experience is only merged if it matches both the task progress and the physical dynamics of an existing record.

E. Part 3: In-Context Policy Injection

The final stage transforms stored experiences into a structured causal prompt for the foundation model.

a) Phase-Conditioned Retrieval.: We use the current phase p_τ to query the persistent memory for the most relevant interaction evidence. The retrieved set $\mathcal{R}_\tau$ consists of signals whose associated phases match the current progress:

$$\mathcal{R}_\tau = \{e_k \mid (p_k, e_k) \in \mathcal{H}_\tau^{\text{pim}}, \text{top-}K \text{ sim}(p_\tau, p_k)\} \quad (7)$$

b) Causal Prompt Construction.: The memory context $\mathcal{M}_\tau$ is constructed by integrating the task token g and the current phase token p_τ with both short term and long term

Algorithm 1 In-Context Causal Inference with Zeva

Input: Task instruction g, environment $\mathcal{E}$, maximum attempts R, episode horizon T, retrieval size K, and brief-memory window K_b
Input: Frozen Causal Transition Encoder (CTE), Memory Context Encoder F_{mem}, and foundation policy π
Output: Task outcome and updated persistent interaction memory $\mathcal{H}^{\text{pim}}$

Freeze all neural parameters θ
$\mathcal{H}^{\text{pim}} \leftarrow \varnothing$ ▷ Global schema retained across attempts
for $r = 1, \ldots, R$ **do**
  $o_0 \leftarrow \textsc{Reset}(\mathcal{E})$
  $\mathcal{H}^{\text{brief}} \leftarrow \varnothing$ ▷ Local context reset per attempt
  $B_0 \leftarrow F_{\text{init}}(g, s_0)$ ▷ Initialize causal interaction state
  $p_0 \leftarrow P_{\text{phase}}(B_0)$ ▷ Initial phase token
  for $t = 0, \ldots, T-1$ **do**
    $\mathcal{R}_t \leftarrow \textsc{PhaseRetrieval}(p_t, \mathcal{H}^{\text{pim}}, K)$ ▷ Retrieve relevant interaction evidence
    $\mathcal{M}_t \leftarrow F_{\text{mem}}(g, p_t, \mathcal{H}^{\text{brief}}, \mathcal{R}_t)$ ▷ Construct Causal Prompt
    $a_t \sim \pi(\cdot \mid o_t, g, \mathcal{M}_t)$ ▷ Implicit causal inference and action generation
    $(o_{t+1}, \texttt{terminal}) \leftarrow \textsc{Step}(\mathcal{E}, a_t)$
    $B_{t+1} \leftarrow \text{GRU}(B_t, [s_{t+1}; a_t; \Delta s_t])$ ▷ Update interaction state via CTE
    $p_{t+1} \leftarrow P_{\text{phase}}(B_{t+1})$, $e_{t+1} \leftarrow P_{\text{signal}}(B_{t+1})$
    $\mathcal{H}^{\text{brief}} \leftarrow \textsc{UpdateBrief}(\mathcal{H}^{\text{brief}}, e_{t+1}, K_b)$ ▷ Update sliding window
    $\mathcal{H}^{\text{pim}} \leftarrow \textsc{CausalMerge}(\mathcal{H}^{\text{pim}}, p_{t+1}, e_{t+1})$
    if `terminal` **then**
      break
    end if
  end for
  if $\textsc{Success}(\mathcal{E})$ **then**
    return $(\textsc{Success}, r, \mathcal{H}^{\text{pim}})$
  end if
end for
return $(\textsc{Failure}, R, \mathcal{H}^{\text{pim}})$

interaction histories. Specifically, the Memory Context Encoder F_{mem} fuses these components after projecting the history streams through a shared projector P_{proj}:

$$\mathcal{M}_\tau = F_{\text{mem}}\left([g;\ p_\tau;\ P_{\text{proj}}(\mathcal{H}_\tau^{\text{brief}});\ P_{\text{proj}}(\mathcal{R}_\tau)]\right) \tag{8}$$

In this formulation, g and p_τ provide task and phase anchoring, while $P_{\text{proj}}(\mathcal{H}_\tau^{\text{brief}})$ and $P_{\text{proj}}(\mathcal{R}_\tau)$ provide causal feedback from the current and previous attempts. By injecting $\mathcal{M}_\tau$ as a **Causal Prompt**, the frozen foundation model performs **implicit causal inference**. It adapts its actions $\pi(a_\tau \mid o_\tau, \mathcal{M}_\tau)$ based on the success or failure of previous interactions to maximize task success.

F. Inference

During deployment, all parameters remain frozen. The agent updates its causal memory online and the foundation model treats the evolving $\mathcal{M}_\tau$ as a dynamic context for decision making. The complete in-context causal inference procedure is summarized in Algorithm 1.

IV. Experiments

We evaluate Zeva on the RoboCasa365-Atomic5 simulation benchmark and the real-world ChemLab-Evo benchmark. Our experiments are designed to answer three key questions: (1) **Benchmark Performance**: Can Zeva achieve competitive task success and long-horizon progress across simulated and real-world manipulation tasks of increasing complexity? (2) **Post-Deployment In-Context Scaling**: Can accumulated interaction experience and a one-shot human demonstration improve a frozen policy across repeated attempts without parameter updates? (3) **Causal Memory and Cross-Task Generalization**: Do CTE, BIT, and PIM encode useful action-induced interaction signals that contribute to performance improvement and support meaningful cross-task retrieval and transfer?

A. Experimental Setup

1) Episode and Attempt Protocol: We use two evaluation units: **episode** and **attempt**. An episode corresponds to a single task instance defined by its initialization, including the scene layout, object poses, and the robot's starting configuration. A **randomized episode** independently samples this initialization at the beginning of the episode. A **fixed episode** selects the initialization once and holds it unchanged across repeated attempts. An attempt is one continuous policy execution from the initial observation until termination or success. Before each new attempt in a fixed episode, the environment is restored to the same initial state.

The Brief Interaction Trace (BIT) is cleared at the beginning of every attempt, whereas the Persistent Interaction Memory (PIM) is retained across attempts within the same episode. Both memories are cleared before the next episode. An episode terminates immediately after the first successful attempt, and every compared method receives the same maximum retry budget.

2) Simulation Benchmark: We evaluate Zeva on five representative tasks from RoboCasa365, covering diverse kitchen manipulation skills such as articulated-object interaction, object placement, and appliance operation. We refer to this five-task evaluation subset as **RoboCasa365-Atomic5** (**Atomic5**). Each task is evaluated over 50 independently randomized episodes.

3) Real-World Chemical Lab (ChemLab-Evo): To evaluate whether Zeva can improve its manipulation capability through accumulated interaction experience after deployment, we construct **ChemLab-Evo**, a real-world chemical laboratory benchmark using an **ARX** manipulator in an $80\,\text{cm} \times 60\,\text{cm}$ workspace. ChemLab-Evo comprises seven tasks spanning three levels of increasing compositional complexity. Level 1 (Atomic) evaluates individual manipulation primitives through Pick Up Test Tube, Place Beaker, and Pour Water. Level 2 (Short-sequence) introduces short skill compositions through Titration and Prepare Salt Solution. Level 3 (Complex) evaluates long-horizon, multi-stage procedures through Balance Weighing and Extraction. As complexity increases, successful execution shifts from completing a single target primitive to completing all constituent phases in the prescribed order. This hierarchical evaluation allows us to examine not only task-level performance, but also whether accumulated interaction experience improves execution reliability as task complexity increases. The main benchmark and each ablation configura-

TABLE I

REAL-WORLD EVALUATION ON CHEMLAB-EVO. THE LEFT BLOCK REPORTS SUCCESS RATE (SR, %) OVER 20 RANDOMIZED EPISODES, WITH MACRO-AVERAGES FOR EACH DIFFICULTY LEVEL. THE RIGHT BLOCK REPORTS THE AVERAGE PERCENTAGE OF ORDERED TASK STAGES COMPLETED ON THE TWO LONG-HORIZON TASKS. BEST PROCESS SCORES ARE IN **BOLD**.

Method	**Success Rate (SR, %)**										**Process Score (%)**		
	Level 1 (Atomic)				**Level 2 (Short-sequence)**			**Level 3 (Complex)**			**Long-horizon**		
	Pick Up Test Tube	**Place Beaker**	**Pour Water**	**Avg. (%)**	**Titration**	**Prepare Salt Solution**	**Avg. (%)**	**Balance Weighing**	**Extraction**	**Avg.**	**Balance Weighing**	**Extraction**	**Avg.**
LingBot-VA [8]	80	70	65	71.7	55	75	65.0	5	0	2.5	28.75	40.00	34.38
Fast-WAM [37]	85	70	75	76.7	45	55	50.0	0	0	0.0	22.50	58.57	40.54
$\pi_{0.5}$ [14]	95	60	65	73.3	60	50	55.0	0	0	0.0	33.75	31.43	32.59
Cosmos3-Nano [18]	80	70	60	70.0	45	60	52.5	0	0	0.0	38.75	50.71	44.73
Zeva	100	70	80	**83.3**	70	70	**70.0**	5	10	**7.5**	**47.50**	**67.14**	**57.32**

TABLE II

RESULTS ON THE ROBOCASA365-ATOMIC5 (ATOMIC5) BENCHMARK [38]. WE REPORT TASK SUCCESS RATE (SR) OVER 50 RANDOMIZED EPISODES AND THE MACRO-AVERAGE. BEST RESULTS ARE IN **BOLD**.

Method	**TurnOn ElectricKettle**	**CloseToaster OvenDoor**	**TurnOn Microwave**	**CoffeeSetup Mug**	**OpenStand MixerHead**	**Avg.**
LingBot-VA [8]	63	60	70	35	48	55.2
Xiaomi-Robotics-1 [39]	17	16	82	6	35	31.2
τ_0-WM [40]	15	32	63	12	23	29.0
Fast-WAM [37]	**88**	66	60	**54**	**94**	72.4
Cosmos3-Nano [18]	76	74	76	6	92	64.8
Zeva	78	**86**	**84**	44	92	**76.8**

tion use 20 independently randomized episodes per evaluated task.

4) Comparative Baselines: On Atomic5, we compare Zeva with LingBot-VA [8], Xiaomi-Robotics-1 [39], τ_0-WM [40], Fast-WAM [37], and Cosmos3-Nano [18]. For ChemLab-Evo, we additionally include $\pi_{0.5}$ [14]. All baselines are reproduced locally using the same task-specific training datasets, observation and action interfaces, and evaluation protocol as Zeva.

B. Evaluation Metrics

Benchmark Metrics. For both simulation and ChemLab-Evo, we report the task-level **Success Rate (SR)** and the macro-average SR across the evaluated tasks. Each SR is computed over 50 randomized episodes per simulation task and 20 randomized episodes per real-world task. Because binary success provides limited resolution for long-horizon tasks, we additionally report a normalized **process Score**. For a task with M_t ordered stages, let L_i be the length of the longest stage prefix completed in the prescribed order during episode i. The task-level Score is

$$\text{Score}_t = \frac{100}{N} \sum_{i=1}^{N} \frac{L_i}{M_t},$$

where $N = 20$ for ChemLab-Evo. An episode receives 100 only if all stages are completed, while a failed episode retains credit for the valid progress made before termination.

Post-Deployment Scaling Metrics. To quantify how capability scales with accumulated interaction experience, let $y_{i,k} \in \{0, 1\}$ denote the outcome of fixed episode i at attempt k. We report the **Cumulative Success Rate (CSR@K)**, the fraction of fixed episodes completed within an attempt budget K:

$$CSR@K = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\max_{1 \le k \le K} y_{i,k} = 1 \right].$$

C. Experiment Results

1) Real-World Evaluation on ChemLab-Evo: We further evaluate Zeva on the real-world ChemLab-Evo benchmark using an ARX manipulator. The SR block of Table I is organized by task complexity, from atomic manipulation primitives to multi-stage chemical procedures. Zeva achieves the best average success rate at every complexity level. Relative to the strongest baseline in each level, it improves the atomic, short-sequence, and complex averages by 6.6, 5.0, and 5.0 percentage points, respectively. Since terminal success alone obscures partial progress on the two long-horizon tasks, the process-score block on the right further compares their ordered stage completion. Zeva obtains Scores of 47.50 on Balance Weighing and 67.14 on Extraction, exceeding the strongest baseline for each task by 8.75 and 8.57 points, respectively. Its average Score of 57.32 improves over the best baseline average by 12.59 points.

2) Simulation Benchmark Results: As shown in Table II, Zeva achieves the highest average success rate of 76.8%, exceeding Fast-WAM by 4.4 percentage points.

D. Ablation Studies

We ablate the two interaction-memory timescales on five real-world ChemLab-Evo tasks: all three atomic tasks and both short-sequence tasks. Table III compares the full model with variants that remove the Brief Interaction Trace, the Persistent Interaction Memory, or both. We denote these components as BIT and PIM, respectively, and report the task-level SR over 20 randomized episodes for each configuration.

The full model performs best on all five tasks. Removing PIM reduces the SR by 10–20 percentage points, while removing BIT causes a larger decrease of 15–30 percentage points, with the largest drops on Pour Water and Titration. These controlled results support the complementary roles of cross-attempt memory and within-attempt context. Disabling both components yields the weakest result on every task.

TABLE III
REAL-WORLD ABLATION OF BIT AND PIM ON FIVE CHEMLAB-EVO TASKS. CHECKMARKS INDICATE ENABLED COMPONENTS. ENTRIES REPORT TASK-LEVEL SR (%) OVER 20 RANDOMIZED EPISODES; BEST RESULTS ARE IN **BOLD**.

Method		Pick Up Test Tube	Place Beaker	Pour Water	Titr.	Salt Solution
BIT	PIM					
✓	✓	**100**	**70**	**80**	**70**	**70**
	✓	80	50	50	40	55
✓		85	60	70	60	50
		65	40	45	25	35

E. Post-Deployment In-Context Scaling

1) Post-Deployment Capability Scaling: We next examine whether Zeva improves its probability of task completion as interaction experience accumulates across repeated attempts. This analysis is conducted on fixed episodes selected separately from the randomized episodes used in the main benchmark tables. For Atomic5, the scaling subset contains 20 fixed episodes per task, yielding 100 equally weighted task–configuration pairs; it is distinct from the standard evaluation above, which uses 50 randomized episodes per task across a broader set of initial configurations. Figure 3 reports four milestones along the evolution trajectory, denoted Evolve 1–4. These labels index selected stages of evolution rather than consecutive attempt numbers. The pooled cumulative success rate rises from 26% at Evolve 1 to 45% at Evolve 2, reaches 70% at Evolve 3, and plateaus at 73% by Evolve 4. On

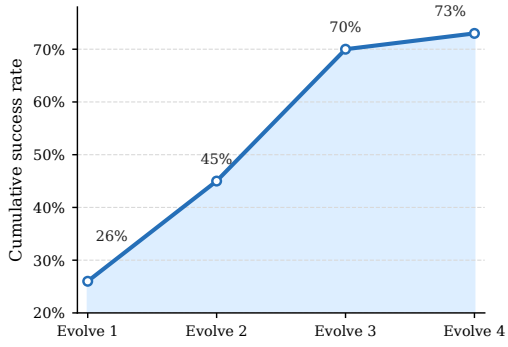


Fig. 3. Post-deployment capability scaling on RoboCasa365. Each of the five tasks contributes 20 randomized episodes, yielding 100 equally weighted task–seed pairs. Evolve 1–4 denote selected evolution milestones.

ChemLab-Evo, Figure 4 likewise uses separately selected fixed episodes and reports four evolution milestones for the three atomic tasks. The plotted fixed episodes are independent of the randomized episodes in Table I. Across these milestones, cumulative success increases monotonically from 65% to 100% for Pick Up Test Tube, from 25% to 70% for Place Beaker, and from 30% to 80% for Pour Water.

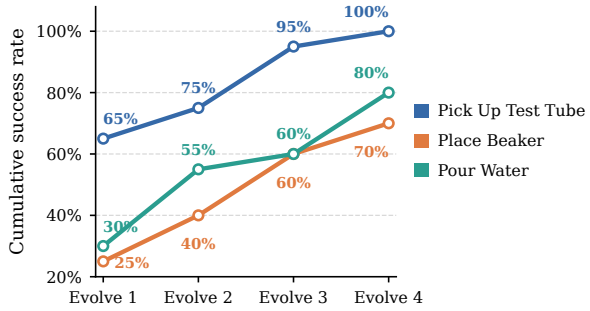


Fig. 4. Post-deployment capability scaling on three ChemLab-Evo tasks using separately selected fixed episodes. Evolve 1–4 denote selected evolution milestones.

2) One-Shot Warm-Up from a Human Demonstration: We further present a one-shot memory warm-up case study on the Balance Weighing task. A human physically guides the ARX manipulator once to pick a calibration weight and place it on the balance; CTE encodes the resulting action–state-change trajectory to initialize PIM while the policy remains frozen. We compare the resulting Zeva execution with a baseline that uses the same frozen policy without memory warm-up. As shown in Figure 5, Zeva reproduces the demonstrated approach, grasp, transfer, and placement stages. Its end-effector trajectory reaches both the grasp and placement events, whereas the baseline deviates after approaching the calibration weights and does not complete a valid placement. This case illustrates that a single human experience can warm-start Zeva through its memory interface without gradient updates before autonomous interaction continues.

We further quantify this warm-up effect on the three atomic ChemLab-Evo tasks. For every evaluated fixed episode, the warm-up condition provides one task-matched human demonstration before autonomous self-evolution, while the comparison condition self-evolves without this initialization. As shown in Figure 6, human warm-up improves or matches SR at every evolution milestone. The gain reaches 20 percentage points on Place Beaker and 15 points on Pour Water; on Pick Up Test Tube, warm-up improves the early milestones before both conditions converge to 100%.

F. Cross-Task Generalization of Causal Interaction Signals

1) Cross-Task Transfer: We first test whether task-local causal interaction signals can be replaced by functionally similar signals retrieved from other tasks during policy execution. For each retrieval, we substitute the corresponding task-local interaction signal with its nearest cross-task match while keeping the policy frozen and all remaining inputs unchanged. As shown in Figure 7, SR changes from 100% to 95% on Pick Up Test Tube and remains 80% on Pour Water. In contrast, replacing the task-local signal with a randomly

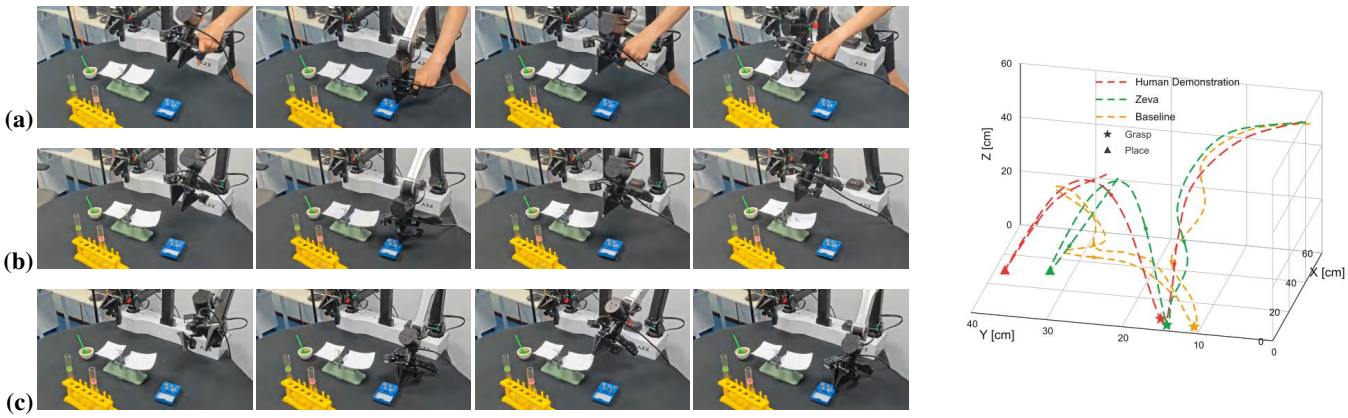


Fig. 5. One-shot human warm-up for Balance Weighing. Rows show (a) the human-guided demonstration used to initialize PIM, (b) Zeva after one-shot memory warm-up, and (c) the baseline without memory warm-up. Frames in each row form a timeline ordered from left to right. The 3D plot compares the corresponding end-effector trajectories; stars and triangles denote grasp and placement events, respectively.

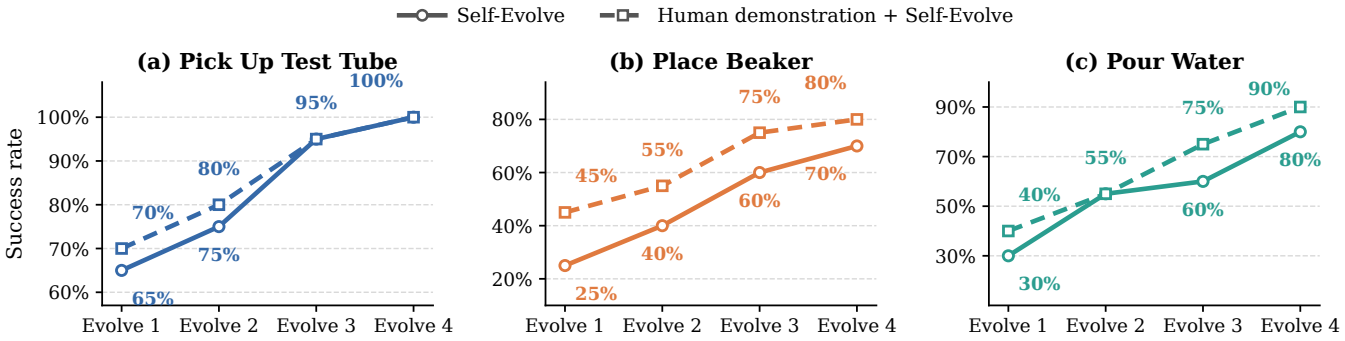


Fig. 6. One-shot human-demonstration warm-up on three atomic ChemLab-Evo tasks. Each evaluated fixed episode is paired with one task-matched human demonstration before autonomous self-evolution. Solid lines show self-evolution without warm-up, while dashed lines show human warm-up followed by self-evolution. Each panel uses its own y-axis range to keep within-task changes legible.

selected cross-task interaction signal reduces SR to 55% and 45%, respectively. Nearest cross-task retrieval therefore exceeds random replacement by 40 and 35 percentage points while preserving most or all of the original task performance in these two settings.

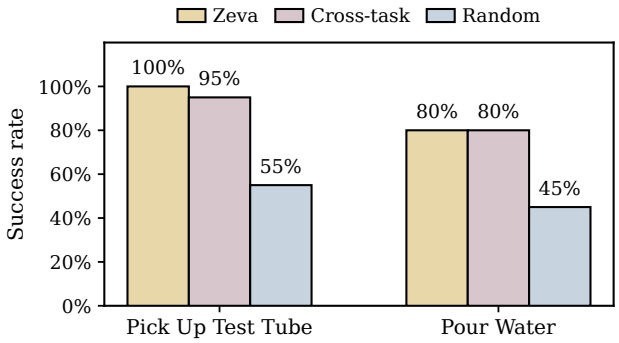


Fig. 7. Quantitative replacement with cross-task causal interaction signals on ChemLab-Evo. Zeva uses the retrieved task-local interaction signal, Cross-task substitutes its nearest counterpart from another task, and Random uses a randomly selected interaction signal from the same cross-task pool. The frozen policy and all other inputs remain unchanged.

2) Cross-Task Retrieval: We next probe whether the CTE causal interaction signal e_τ captures action-induced state changes beyond task-specific appearance. For each query transition, we retrieve nearest neighbors by cosine similarity between interaction signals from a diagnostic memory pool that contains only other tasks; this analysis does not alter the task-local memory protocol used for the quantitative evaluation. Figure 8 shows representative matches for three effects.

A pouring transition from Pour Water retrieves container-tilting transitions from Prepare Salt Solution and Extraction. Gripper closure around a test tube retrieves contact-establishment transitions involving different vessels, while lifting retrieves post-grasp upward motions across object categories. Thus, transitions with different objects, viewpoints, and task instructions are aligned by functionally equivalent physical effects, providing a mechanism for future cross-task memory reuse.

G. Qualitative and Representation Analysis

1) Case-Wise Post-Deployment Evolution: Figure 9 complements the aggregate scaling curves with case-wise evidence of how interaction memory changes later retries and the eventual successful execution. The four frames in each row

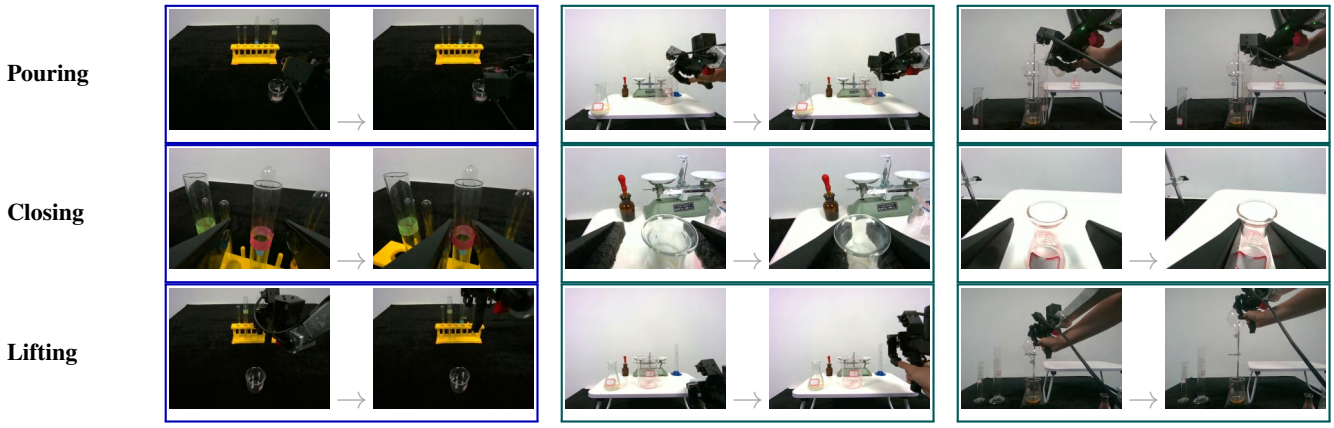


Fig. 8. Cross-task retrieval in the latent space of CTE causal interaction signals. Each block shows a left-to-right state transition. Blue outlines denote queries, and teal outlines denote their two nearest cross-task matches. Rows group transitions by physical effect despite changes in objects, viewpoints, and instructions.

are arranged chronologically, but they do not represent four distinct attempts.

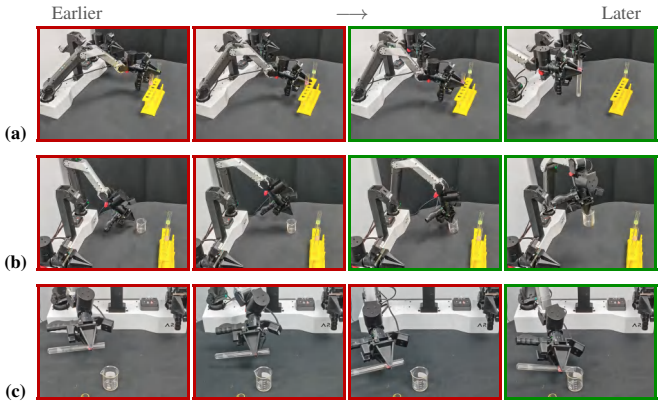


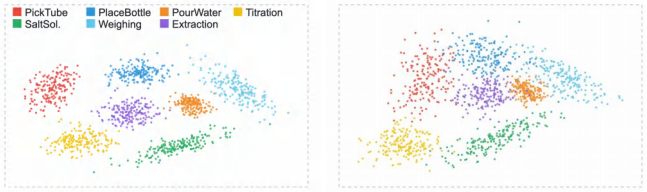
Fig. 9. Chronological post-deployment evolution on three ChemLab-Evo tasks. Red frames are terminal observations from distinct failed attempts. Green frames show the subsequent successful attempt; in (a) and (b), the two green frames are successive stages of the same continuous attempt.

Pick Up Test Tube. In the first two attempts, the gripper reaches the rack but does not establish a stable grasp. In the subsequent successful attempt, the third frame shows the corrected grasp, while the fourth shows the robot retracting with the tube during the same continuous execution. This progression is consistent with the cross-attempt use of the action–effect evidence retained in PIM.

Place Beaker. The first two attempts expose errors in wrist orientation and placement near the target region. The final two frames then show two stages of one successful attempt: Zeva first secures the beaker and subsequently places it at the target. This corrected execution illustrates the type of cross-attempt adaptation supported by the retained interaction evidence.

Pour Water. This task additionally requires coordinated transport, rim alignment, and controlled rotation. Across the first three failed attempts, the observed behavior reveals remaining relative-pose and tilt errors. In the subsequent successful attempt, Zeva positions the tube over the beaker and executes the required pouring motion. This qualitative sequence is consistent with progressive correction from accumulated interaction evidence.

2) CTE Representation Analysis: We further use t-SNE to visualize the interaction embeddings learned by CTE across all seven ChemLab-Evo tasks. Figure 10 compares the full three-stream representation with a variant that removes the effect stream. Full CTE produces compact, task-specific clusters with clear separation, whereas removing the effect stream increases within-task dispersion and inter-task mixing. This comparison is consistent with observed state changes contributing to task-discriminative causal interaction memory.



(a) Full CTE **(b) w/o Effect Stream**

Fig. 10. t-SNE visualization of CTE interaction embeddings across seven ChemLab-Evo tasks. Full CTE forms compact clusters, whereas removing the effect stream increases dispersion and cross-task overlap.

V. Conclusion

Zeva is the first framework to enable in-context learning of interaction causality from a robot's own physical experience for generalizable embodied manipulation, while keeping the policy model frozen. Experiments in simulation and real-world manipulation demonstrate that Zeva achieves the best performance among the frontier VLA and WAM models. Moreover, its success rate consistently improves over repeated attempts as interaction experience accumulates.

A current limitation is that Zeva learns causality only from attempts generated for task execution, rather than actively collecting interactions for learning. Future work will therefore investigate causality-driven exploration, in which the robot purposefully selects informative rollouts to reduce uncertainty about physical interactions and acquire experience more efficiently.

ACKNOWLEDGMENT

We thank An Pan, Zexu Wang, Yi Tao, Zijian Wang, Shuhao Wu, Wenhui Gu, and Bowen Yang for their contributions on engineering and demos.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "OpenVLA: An open-source vision-language-action model," in *Proc. Conf. Robot Learning (CoRL)*, 2024.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," arXiv preprint arXiv:2410.24164, 2024.

[3] A. Brohan, N. Brown, J. Carbajal *et al.*, "RT-1: Robotics transformer for real-world control at scale," arXiv preprint arXiv:2212.06817, 2022.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. Conf. Robot Learning (CoRL)*, 2023.

[5] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[7] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu, "Cosmos policy: Fine-tuning video models for visuomotor control and planning," arXiv preprint arXiv:2601.16163, 2026.

[8] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu, "Causal world modeling for robot control," arXiv preprint arXiv:2601.21998, 2026.

[9] C. Ye, Y. Ge, Y. Ge, Y. Shan, and M.-Y. Liu, "World action models are zero-shot policies," arXiv preprint arXiv:2602.15922, 2026.

[10] Y. Jiang, Y. Chebotar, R. Zheng, F. Hu, Y. Ge, J. Wu, T. Dai, S. Reed, L. Fei-Fei, Y. Zhu, and L. Fan, "RoboTTT: Context scaling for robot policies," arXiv preprint arXiv:2607.15275, 2026.

[11] Y. Feng, B. Han, J. Lyu, K. Liu, Y. Zheng, Y. Wan, W. Liu, S. Han, R. Li, Y. Zhang, F. Liu, X. Shi, L. Liu, Y. Wang, Z. Zhang, and H. Wang, "WAM-TTT: Steering world-action models by watching human play at test time," arXiv preprint arXiv:2607.06988, 2026.

[12] Generalist Team, "GEN-1.5: Embodied foundation models are one-shot learners," Generalist AI Blog, Aug. 2026. [Online]. Available: https://generalistai.com/blog/gen-1.5

[13] Skild AI, "Introducing S1: In-context learning for robotics," Skild AI Blog, Aug. 2026. [Online]. Available: https://www.skild.ai/blogs/s1

[14] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," arXiv preprint arXiv:2504.16054, 2025.

[15] Physical Intelligence *et al.*, "$\pi^*_{0.6}$: a VLA that learns from experience," arXiv preprint arXiv:2511.14759, 2025.

[16] H. Wu, Y. Jing, C.-L. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.

[17] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu, "GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation," arXiv preprint arXiv:2410.06158, 2024.

[18] NVIDIA, "Cosmos 3: Omnimodal world models for physical ai," https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf, 2026.

[19] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[20] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto, "Behavior generation with latent actions," in *Proc. Int. Conf. Machine Learning (ICML)*, 2024.

[21] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: Efficient action tokenization for vision-language-action models," arXiv preprint arXiv:2501.09747, 2025.

[22] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "RT-H: Action hierarchies using language," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[23] R. Bonatti, S. Vemprala, S. Ma, F. Frujeri, S. Chen, and A. Kapoor, "PACT: Perception-action causal transformer for autoregressive robotics pre-training," arXiv preprint arXiv:2209.11133, 2022.

[24] Y. Zhou, Q. Liang, S. Zhuang, J. Li, X. Wang, B. Cai, Y. Mo, and R. Xu, "Action-effect memory pretraining for robot manipulation," arXiv preprint arXiv:2606.12499, 2026.

[25] Z. Tang, H. Liu, X. Chang, C. Wu, D. Huo, Y. Yang, B. Liu, Z. Cai, F. Xiong, M. Xu, J. Luo, D. Ma, Z. Ma, and G. Pan, "ALAM: Algebraically consistent latent transitions for vision-language-action models," arXiv preprint arXiv:2605.10819, 2026.

[26] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo, "Latent action pretraining from videos," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2025.

[27] B. Hu, Z. Li, R. Shao, J. Chen, A. H. Liu, W.-S. Zheng, and L. Nie, "From abstraction to instantiation: Learning behavioral representation for vision-language-action model," in *Proc. Int. Conf. Machine Learning (ICML)*, 2026.

[28] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, "UniVLA: Learning to act anywhere with task-centric latent actions," arXiv preprint arXiv:2505.06111, 2025.

[29] H. Shi, B. Xie, Y. Liu, L. Sun, F. Liu, T. Wang, E. Zhou, H. Fan, X. Zhang, and G. Huang, "MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2026.

[30] M. Torne, K. Pertsch, H. Walke, K. Vedder, S. Nair, B. Ichter, A. Z. Ren, H. Wang, J. Tang, K. Stachowicz, K. Dhabalia, M. Equi, Q. Vuong, J. T. Springenberg, S. Levine, C. Finn, and D. Driess, "MEM: Multi-scale embodied memory for vision language action models," arXiv preprint arXiv:2603.03596, 2026.

[31] K. Wang, Z. Gu, Y. Chen, Y. Xu, Q. Ma, J. Yang, Z. Li, Y. Huang, L. Wang, and P. Su, "DIM-WAM: World-action modeling with diverse historical event memory," arXiv preprint arXiv:2606.27677, 2026.

[32] S. Yang, J. Mu, T. Wei, C. Lu, X. Li, L. Xu, Z. Xue, Z. Yuan, D. Lin, J. Pang, and H. Xu, "MemoryWAM: Efficient world action modeling with persistent memory," arXiv preprint arXiv:2606.20562, 2026.

[33] G. Lu, W. Guo, C. Zhang, Y. Zhou, H. Jiang, Z. Gao, Y. Tang, and Z. Wang, "VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning," arXiv preprint arXiv:2505.18719, 2025.

[34] X. Ding, L. Mi, M. Huang, Z. Wang, C. Zhang, Z. Hao, F. Chen, X. Li, Y. Zheng, Y. Guo *et al.*, "Zetta

\

zeta : $Anefficientclosed-loopembodiedharnessforself-evolvingphysicalintelligence,''$ *arXiv preprint arXiv:2608.16590*, 2026.

[35] R. Wu, X. Wang, J. Mei, P. Cai, D. Fu, C. Yang, L. Wen, X. Yang, Y. Shen, Y. Wang *et al.*, "Self-evolving llm agents through an experience-driven lifecycle," *arXiv preprint arXiv:2510.16079*, 2025.

[36] J. Zhang, S. Hu, C. Lu, R. Lange, and J. Clune, "Darwin gödel machine: open-ended evolution of self-improving agents," in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 104 223–104 294.

[37] T. Yuan, Z. Dong, Y. Liu, and H. Zhao, "Fast-WAM: Do world action models need test-time future imagination?" arXiv preprint arXiv:2603.16666, 2026.

[38] S. Nasiriany, S. Nasiriany, A. Maddukuri, and Y. Zhu, "RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2026.

[39] Xiaomi Robotics Team, J. Guo, P. Jin, J. Li, P. Li, Y. Li, F. Liu, W. Peng, O. Qin, Y. Su, N. Sun, Q. Sun, R. Suo, H. Wang, Y. Wang, R. Wu, C. Xia, L. Zhang, J. Zhao, G. Chen, W. Chen, X. He, B. Li, Q. Li, Z. Li, H. Qu, W. Song, D. Xiang, Y. Xie, P. Xu, H. Ye, W. Ye, H. Zhao, and Q. Zhou, "Xiaomi-Robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories," arXiv preprint arXiv:2607.15330, 2026.

[40] P. Zhou, S. Chen, D. Chen, J. Wang, R. Jin, B. Zhu, Y. Pan, S. Gu, K. Wang, S. Nan, X. Qiu, C. Qiu, P. Yang, Y. Cai, J. Gao, Y. Li, Y. Fu, X. Yue, Z. Chen, and J. Luo, "τ_0-WM: A unified video-action world model for robotic manipulation," arXiv preprint arXiv:2606.01027, 2026.