WORLD ACTION MODELS ENABLE CONTINUAL IMITATION LEARNING WITH GENERATIVE REPLAYS

Anonymous authors
Paper under double-blind review

ABSTRACT

Going beyond predicting robot actions, World Action Models (WAMs) can also generate future visual observations. We build on this generative capability to propose Recurrent Generative Replay (REGEN), a continual imitation learning framework that synthesizes pseudo-replay trajectories, enabling a robot policy to rehearse previously learned tasks without storing their original human demonstrations. During continual adaptation, REGEN recursively queries the WAM to synthesize pseudo-replay trajectories conditioned only on prior task instructions and current-task observations. Experiments in both simulation and real-world manipulation settings show that REGEN reduces catastrophic forgetting over 50% relative to sequential fine-tuning, while approaching the performance of privileged experience replay methods that require access to real replay data. Finally, we analyze the factors limiting generated replay, identifying long-horizon visual degradation and action-observation inconsistency as the primary bottlenecks. Our results establish WAMs as a promising foundation for continual robot learning without stored demonstrations.

"A robot that can imagine its past can continue learning its future."

1 INTRODUCTION

Recently, World Action Models (WAMs) have emerged as a promising paradigm for robot imitation learning by unifying *perception*, *prediction*, and *control* within a single generative framework Ye et al. (2026b); Kim et al. (2026); Li et al. (2026); Ye et al. (2026a). As WAMs become general-purpose robot policies, continual adaptation to new tasks becomes inevitable. Yet, like other learned policies, they suffer from catastrophic forgetting when fine-tuned sequentially French (1999); Luo et al. (2025); Shenfeld et al. (2026). This raises a key question: *can the same generative capabilities that make WAMs powerful also serve as a mechanism for retaining previously learned skills?*

Existing approaches for mitigating catastrophic forgetting span two generations of robot policies: conventional visuomotor policies Chi et al. (2024); Zhao et al. (2023) and more recent Vision-Language-Action (VLA) models Kim et al. (2024); Black et al. (2024); Intelligence et al. (2025). Across both paradigms, experience replay has emerged as the dominant continual learning strategy, where demonstrations from previous tasks are stored and replayed during adaptation to new tasks Liu et al. (2023); Zhu et al. (2022); Wan et al. (2024); Liu et al. (2026b); Wu et al. (2026). While highly effective, these approaches fundamentally rely on access to ground-truth demonstrations from prior tasks. This assumption is becoming increasingly impractical in modern robot learning. Earlier progress in robotics was driven by open datasets of robot demonstrations Collaboration et al. (2023); Bu et al. (2025a); Tian et al. (2025), whereas the emerging paradigm is centered around pretrained robot foundation models trained on large-scale proprietary datasets that are rarely released publicly Black et al. (2024); Intelligence et al. (2025); Ye et al. (2026b). Consequently, what we desire for continual learning is a replay mechanism that does not depend on access to *true* demonstrations from previous tasks.

WAMs are uniquely positioned to address exactly this challenge. By jointly modeling actions together with future visual observations of the scene Ye et al. (2026b); Kim et al. (2026); Li et al. (2026); Ye et al. (2026a), WAMs expose a generative interface capable of synthesizing trajectories for previously learned tasks conditioned only on their language instructions and current observations. In this work, we leverage this property to enable continual imitation learning *without* access to any real demonstrations from previous tasks as shown in Figure 1.

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

Consequently, we propose **Re**current **Gen**erative Replay (**REGEN**), the first continual learning framework that leverages the WAM itself as a native generative replay mechanism. When adapting to a new task, REGEN generates pseudo-demonstrations for previous tasks by conditioning the WAM on prior task instructions and initial visual observations, then recurrently feeding back its own generated future observations. This yields completed synthetic trajectories of previous tasks. These pseudo-demonstrations are then combined with the new-task demonstrations during fine-tuning to substantially mitigate catastrophic forgetting without requiring human-collected demonstrations of past tasks. We evaluate REGEN in both simulated Liu et al. (2023) and real-world environments, where it reduces forgetting by over 50% relative to naive sequential fine-tuning and approaches the performance of experience replay methods that rely on privileged access to human-collected demonstrations from previous tasks. Critically, our analysis reveals that the effectiveness of REGEN is fundamentally tied to two key limitations of current WAMs: **(i)** degradation in the visual fidelity of future observations during long-horizon recurrent generation, and **(ii)** inconsistencies between predicted future observations and the corresponding generated actions. These findings indicate that, while REGEN is not yet a complete substitute for real replay, the remaining performance gap is primarily governed by the generative limitations of WAMs themselves. Consequently, improving the fidelity and consistency of WAM generation represents a key direction for closing the gap between pseudo-replay and real experience replay.

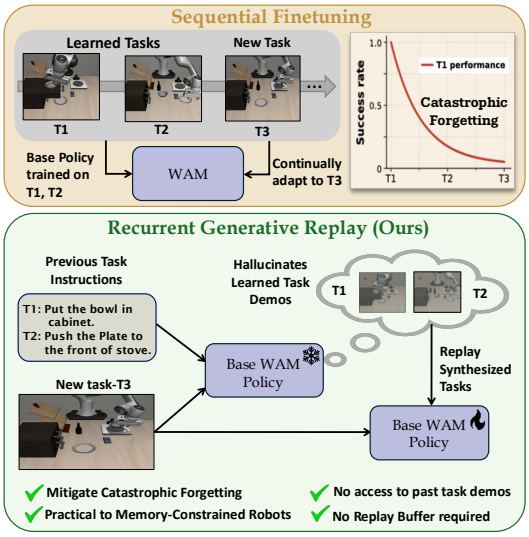


Figure 1: **Comparison of REGEN and Sequential-Finetuning.** **Top:** Sequential fine-tuning of WAMs leads to catastrophic forgetting. **Bottom:** REGEN leverages the WAM's generative capabilities to hallucinate pseudo-demonstrations of previously learned tasks, replaying them alongside new task data to mitigate forgetting without storing any prior-task demonstrations.

We summarize our contributions as follows:

- We propose **Recurrent Generative Replay**, the first replay-based continual imitation learning framework that leverages the generative capabilities of WAMs to synthesize replay trajectories without storing real demonstrations from previous tasks.
- We demonstrate that REGEN substantially mitigates catastrophic forgetting in both simulated and real-world manipulation settings while preserving strong forward transfer.
- We identify the primary limitations preventing generated replay from matching real experience replay, namely long-horizon degradation in visual generation and inconsistencies between predicted observations and actions, highlighting key future directions for WAMs.

2 Related Work

Continual Imitation Learning. Continual learning aims to acquire a sequence of tasks without catastrophically forgetting previously learned skills (French, 1999). Existing approaches fall into three families: *regularization-based* methods that constrain weight updates (Kirkpatrick et al., 2017; Zenke et al., 2017), *rehearsal-based* methods that retain real samples from prior tasks (Chaudhry et al., 2019), and *architecture-based* methods that allocate task-specific parameters (Mallya & Lazebnik, 2018; Rusu et al., 2016). Prior work (Liu et al., 2023) adapts these to lifelong robot manipulation, and recent methods extend them with skill discovery, distillation, or task-specific adapters combined with replay (Wan et al., 2024; Roy et al., 2025; Liu et al., 2024; Römer et al., 2026); a parallel line of work develops continual learning for VLAs using replay (Wu et al., 2026; Liu et al., 2026a;b). Across this line of work, forgetting is mitigated by storing real demonstrations,

regularizing weights, or learning task-specific adapters. REGEN is orthogonal: rather than maintaining a growing buffer of real trajectories, it leverages the world-action model itself to simulate past tasks, providing replay examples without retaining any real demonstrations.

Generative Replay. Unlike replay-based methods, generative replay synthesizes pseudo-samples for rehearsal (Shin et al., 2017). In robotics, CRIL (Gao et al., 2021), t-DGR (Yue et al., 2024), and (Pan et al., 2025) train generative models on past data to prevent forgetting. In contrast, REGEN generates trajectories from current task data to facilitate continual learning. REGEN shares this current-data insight but targets continual imitation learning. By leveraging a unified world-action model (Kim et al., 2026) for both action and future-frame prediction, REGEN avoids model-based planning and eliminates the need for separate trajectory generator and action policy.

World Models for Robot Control. World models learn predictive representations of environment dynamics, originally proposed for sample-efficient reinforcement learning (RL) (Ha & Schmidhuber, 2018) and popularized by the Dreamer framework (Hafner et al., 2020; 2024). While traditionally used for RL, world models have recently been scaled through video generation backbones (NVIDIA, 2025; Wan et al., 2025), enabling a new class of world action models (WAMs) for robot learning (Kim et al., 2026; Li et al., 2026; AgiBot-World-Contributors et al., 2025; Ye et al., 2026b; Zhu et al., 2025; Team et al., 2026). Unlike world-model-based policies, which generate future observations and rely on a separate policy for actions (Guo et al., 2025; Pai et al., 2025), WAMs predict both jointly. Prior work employs WAMs for control, planning, and synthetic data generation. In contrast, we use the WAM's generative interface as a memory of its own prior tasks, enabling continual learning without stored demonstrations.

3 PRELIMINARIES: WORLD ACTION MODELS

We consider robotic policies learned through imitation learning from expert demonstrations. For each task $\mathcal{T}^k$, specified by a natural language instruction ℓ^k, we assume access to a distribution of demonstrations $\mathcal{D}^k = \{(\ell^k, \tau_i^k)\}_{i=1}^{N_k}$, where each trajectory is defined as $\tau = \{(\mathbf{o}_t, \mathbf{a}_t)\}_{t=1}^{T}$ with observation $\mathbf{o}_t$ and action $\mathbf{a}_t$ at time step t. Each observation consists of multi-view RGB images $\mathbf{I}_t^1, \ldots, \mathbf{I}_t^n$ and the robot proprioceptive state $\mathbf{q}_t$.

WAMs are policies that jointly model future actions and future visual observations conditioned on language instructions and current observation. Unlike conventional policies that predict only actions, WAMs build upon generative video foundation models (NVIDIA, 2025; Wan et al., 2025; Seedance et al., 2026; Zheng et al., 2026), enabling both control and explicit prediction of future scene dynamics. Formally, at each time step t, a WAM parameterized by θ models the joint conditional distribution

$$(\tilde{\mathbf{a}}_{t:t+H},\ \tilde{\mathbf{o}}_{t+H},\ \tilde{r}_t) \sim \pi_\theta\left(\cdot \mid \mathbf{o}_t, \ell\right), \tag{1}$$

where $\tilde{\mathbf{a}}_{t:t+H}$ denotes a predicted action chunk of horizon H, $\tilde{\mathbf{o}}_{t+H}$ is the predicted future observation, and $\tilde{r}_t \in [0, 1]$ estimates task progress. Specifically, $\tilde{r}_t$ predicts the terminal reward $R(\mathbf{o}_T, \mathbf{a}_T)$ from the current state and serves as a dense proxy for proximity to task completion. The model parameters θ are optimized over $\mathcal{D}$ using a combination of behavioral cloning losses for actions, generative objectives for future observations, and regression losses for reward prediction.

4 METHOD

In this section, we first present the continual learning problem formulation and then introduce our proposed framework for continual adaptation of WAMs, **Recurrent Generative Replay** (**REGEN**).

4.1 PROBLEM FORMULATION

We consider continual adaptation of a pretrained WAM policy. Specifically, let π_{θ_0} denote a base policy trained on a set of previously learned tasks $\mathcal{T}_{\text{prev}} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$. Our goal is to adapt π_{θ_0} to a novel task $\mathcal{T}_k$, where $k > M$, while preserving performance on all tasks in $\mathcal{T}_{\text{prev}}$. This results in an updated policy π_{θ_k}.

For each previous task $\mathcal{T}_i \in \mathcal{T}_{\text{prev}}$, we assume access only to the task-level language instruction ℓ_i; no action-observation trajectories from previous tasks are retained. In contrast, for the current task $\mathcal{T}_k$, we assume access to both the task instruction ℓ_k and a distribution of expert demonstrations

$\mathcal{D}_k = \{(\ell_k, \tau_k^n)\}_{n=1}^{N_k}$, where τ_k^n denotes the n-th demonstration trajectory for task $\mathcal{T}_k$. Consequently, continual learning must be performed using demonstrations from the current task alone, while relying only on language instructions to preserve previously acquired behaviors.

A straightforward approach is to fine-tune the policy solely on $\mathcal{D}_k$; however, without supervision from previous tasks, this can lead to catastrophic forgetting. Replay-based continual learning addresses this issue by jointly training on prior and current-task experiences, which is unavailable in our setting. We therefore first introduce **Rollouts-as-Replay (RAR)**, a conceptually simple yet practically constrained alternative for recovering prior-task experience, before presenting our proposed **REGEN** framework for continual policy learning without access to past demonstrations.

4.2 ROLLOUTS AS REPLAY

First, we describe RAR, a simulator-grounded oracle for replay. For each prior task $\mathcal{T}_i$, the base WAM policy π_{θ_0} is conditioned on its instruction ℓ_i and rolled out closed-loop in the corresponding environment. At each step, the policy predicts an action chunk $\tilde{\mathbf{a}}_{t:t+H}$, which is executed in the simulator to obtain the subsequent observation $\mathbf{o}_{t+H}$. The resulting observation-action pairs form the replay set $\mathcal{R}_k$, which is jointly trained with the current-task demonstrations $\mathcal{D}_k$. By using environment-rendered observations, RAR provides a reference for replay with grounded visual feedback. However, it requires access to and repeated interaction with prior-task environments, making it impractical for real-world continual learning, where reproducing previous scenes, executing partially forgotten policies, and supervising physical rollouts can be costly and unsafe. This motivates REGEN, which eliminates environment interaction by leveraging the WAM's generative capability to synthesize prior-task trajectories.

4.3 RECURRENT GENERATIVE REPLAY

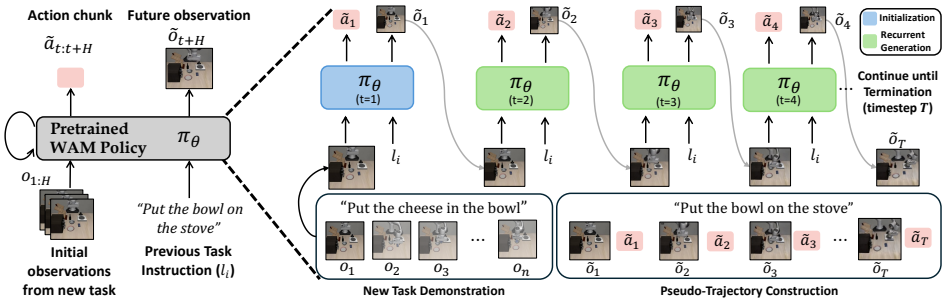


Figure 2: **Overview of the pseudo-trajectory generation process.** *Left:* unrolled view of the REGEN. *Right:* REGEN rolls out π_θ recurrently to construct a pseudo-trajectory for a previous task. The policy is seeded with new-task observations and the previous task's instruction ℓ_i (initialization, blue), then each generated observation is fed back to produce the next $(\tilde{o}_t, \tilde{a}_t)$ pair (recurrent generation, green) until timestep T. For simplicity, we set $H = 1$.

Given the current task $\mathcal{T}_k$ with demonstrations $\mathcal{D}_k$ and a pretrained policy π_θ, REGEN generates pseudo-demonstrations for each previous task $\mathcal{T}_i$ $(i \leq M)$ by conditioning π_θ on the corresponding task instruction ℓ_i and initializing the rollout from a real observation sampled from $\mathcal{T}_k$, as illustrated in Figure 2. At each rollout step, the model predicts an action chunk together with a future observation, which are recurrently fed back into the model to synthesize a pseudo-demonstration trajectory. Overall, pseudo-demonstration generation proceeds in three stages: initialization, recurrent generation, and termination, described below.

Initialization Phase $(0 \leq t < H)$**.** The rollout is initialized using one full action chunk from the current-task demonstrations, since at least one real observation context is required before recursive generation can begin. During this phase, the model conditions on real observations sampled from $\mathcal{D}_k$, i.e., $\mathbf{o}_t^{\text{in}} = \mathbf{o}_t$ for $0 \leq t < H$.

Recurrent Generation Phase $(H \leq t \leq T_{\max})$**.** After initialization, real observations are no longer used. Instead, the model recursively conditions on its own previously generated observations to

produce a fully generative rollout:

$$\mathbf{o}_t^{\text{in}} = \begin{cases} \mathbf{o}_t, & 0 \le t < H, \\ \tilde{\mathbf{o}}_t \;\; \text{where} \;\; (\tilde{\mathbf{a}}_{t-H:t},\, \tilde{\mathbf{o}}_t) \sim \pi_\theta\big(\cdot \mid \mathbf{o}_{t-H}^{\text{in}},\, \ell_i\big), & H \le t \le T_{\max}. \end{cases} \tag{2}$$

This recursive feedback of generated observations induces a recurrent rollout process, enabling reconstruction of prior-task trajectories without storing any previous task demonstrations.

Termination. Trajectory generation terminates either at a maximum horizon $T_{\max}$ or earlier when the goal-reward head consistently predicts task completion. Specifically, generation stops when the predicted goal reward $\tilde{r}_t$ exceeds 0.99 for three consecutive rollout steps and attains 1.0 at least once within that interval.

Pseudo-Trajectory Construction. At each rollout step, the input observation is paired with the first action of the predicted action chunk to construct a pseudo-trajectory:

$$\tilde{\tau}^{\,i} = \big\{\big(\mathbf{o}_t^{\text{in}}, \tilde{\mathbf{a}}_t\big)\big\}_{t=0}^{T_i}, \tag{3}$$

where $\tilde{\mathbf{a}}_t$ denotes the first action of the chunk predicted at time step t, and $T_i \le T_{\max}$ is the generated trajectory length for task $\mathcal{T}_i$. The pseudo-demonstration set is formed by aggregating generated rollouts across all previous tasks, i.e., $\mathcal{R}_k = \bigcup_{i=1}^{M} \tilde{\tau}^{\,i}$. The policy is then updated via behavioral cloning using the combined training set $\mathcal{D}_k^+ = \mathcal{D}_k \cup \mathcal{R}_k$.

Continual Training Objective. The WAM policy is trained on the union of current-task demonstrations and replayed pseudo-trajectories, $\mathcal{D}_k^+$, in order to acquire the new task while retaining performance on previous tasks. Since the WAM encodes the predicted action chunk, future observation, and reward jointly as latent frames $\mathbf{z} = [\mathbf{z}^a, \mathbf{z}^o, \mathbf{z}^r]$, all three are supervised by a single denoising objective:

$$\min_\theta \; \mathbb{E}_{(\mathbf{z},\, \mathbf{o}_t,\, \ell)\sim\mathcal{D}_k^+,\, \tau,\, \boldsymbol{\epsilon}} \Big[\big\| v_\theta(\mathbf{z}^\tau,\, \tau \mid \mathbf{o}_t,\, \ell) - (\boldsymbol{\epsilon} - \mathbf{z})\big\|^2\Big], \tag{4}$$

where $\mathbf{z}^\tau = \tau\boldsymbol{\epsilon} + (1-\tau)\mathbf{z}$ is the noised latent at flow time $\tau \sim \mathcal{U}(0,1)$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and v_θ is the predicted velocity field. Along with policy learning, future observation prediction is necessary for the model to adapt its visual predictions to each new task, so that it enables replay generation for the following stages. The task instruction ℓ corresponds to ℓ_k for samples from the current-task dataset $\mathcal{D}_k$, and to ℓ_i for pseudo-trajectories generated for previous task $\mathcal{T}_i$. In this way, REGEN mitigates catastrophic forgetting by approximating the trajectory distributions of previous tasks without storing ground-truth demonstrations.

5 EXPERIMENTS

We conduct experiments in both simulation and real-world environments to evaluate the effectiveness of the pseudo-trajectories generated by REGEN. We further perform representation analyses to characterize the quality of the generated trajectories and assess their deviation from perfect demonstrations.

5.1 IMPLEMENTATION DETAILS & EVALUATION METRICS

We instantiate Cosmos-Policy (Kim et al., 2026) as the underlying WAM backbone, initialized from scratch. The model conditions on third-person RGB and wrist-camera observations, the current proprioceptive state, and a language instruction, and jointly predicts an action chunk of horizon $H = 16$ at each timestep. Unless otherwise specified, we adopt the training hyperparameters of Cosmos-Policy. The base policy is trained for 10K iterations, and each continual learning stage finetunes the policy for 2K iterations from the previous stage model checkpoint. For replay generation, we synthesize 10 pseudo-trajectories per previous task in $\mathcal{T}_{\text{prev}}$.

We report three standard continual learning metrics computed from task success rates: Forward Transfer (FWT) (Liu et al., 2023; Wan et al., 2024), which measures the ability to acquire new tasks, Negative Backward Transfer (NBT) (Liu et al., 2026a), which quantifies forgetting of previously learned tasks, and Area Under the Curve (AUC) (Liu et al., 2023; Wan et al., 2024). Additional architectural details, training hyperparameters, and evaluation protocol are provided in the Appendix A.

5.2 SIMULATION EXPERIMENTS

Setting. We conduct our simulation experiments on LIBERO benchmark (Liu et al., 2023). We use three different task suites that capture distinct forms of distribution shift:(LIBERO-Spatial, LIBERO-Object, LIBERO-Goal). Each suite contains 10 tasks, with 50 high-quality human-teleoperated demonstrations per task. We follow the continual learning setup of prior work (Wan et al., 2024; Yu et al., 2026), which involves two stages: (1) a base stage and (2) a continual learning stage. In the base stage, six tasks are used for pretraining. During the continual learning stage, the remaining four tasks are introduced sequentially, one per stage. We adopt the task ordering defined for each benchmark suite in (Liu et al., 2026a).

Baseline Methods. We compare REGEN against the following continual learning approaches: **Sequential Fine-Tuning (Seq-FT)** (Liu et al., 2023): fine-tunes the base policy on each new task. **Sequential LoRA (Seq-LoRA)**: a parameter-efficient variant of Seq-FT using Low-rank adaptation (Hu et al., 2022) instead of full parameter updates. **Elastic Weight Consolidation (EWC)** (Kirkpatrick et al., 2017): a regularization based approach which penalizes the parameter updates important for prior tasks. **PackNet** (Mallya & Lazebnik, 2018): an iterative pruning approach that frees up redundant parameters after training each task and uses them to learn new tasks, keeping previously allocated parameters frozen. **Experience Replay (ER)** (Liu et al., 2023): a replay-based method that stores real demonstrations of prior tasks and mixes them with current-task data during training. We include ER only as an upper-bound reference, since it violates our no-real-data assumption.

Table 1: Comparison of REGEN with continual learning methods on LIBERO. *Requires stored demonstrations (ER) or access to prior-task environment (RAR) which are reported as upper-bound references. **Bold** indicates the best results among methods operating in our setting.

	LIBERO-Object			LIBERO-Goal		
Method	FWT ↑	NBT ↓	AUC ↑	FWT ↑	NBT ↓	AUC ↑
Seq-FT (Liu et al., 2023)	92.7	82.6	24.9	90.6	100	10.3
Seq-LoRA (Lee et al., 2024)	93.5	99.9	11.1	83.0	99.6	9.3
EWC (Kirkpatrick et al., 2017)	94.9	87.5	25.1	83.6	99.4	9.9
PackNet (Mallya & Lazebnik, 2018)	**95.5**	99.7	11.5	**92.0**	100	10.5
REGEN (ours)	95.3	**26.1**	**65.5**	90.6	**44.9**	**40.8**
ER* (Chaudhry et al., 2019)	95.7	4.8	93.4	94.0	7.2	92.4
RAR* (ours)	96.9	3.0	95.2	92.8	15.4	82.6

Results. Table 1 reports continual learning performance on the two LIBERO benchmark suites. Firstly, Sequential finetuning (Seq-FT) of WAMs achieves strong forward transfer but suffers from near-complete forgetting of previously learned tasks, consistent with prior observations in behavior cloning approaches (Liu et al., 2023; Wan et al., 2024; Römer et al., 2026). While Seq-FT fully adapts to new tasks but forgets prior ones, the non-replay baselines (Seq-LoRA, EWC, and PackNet) retain partial knowledge of previous tasks but cannot perform them reliably. Next, ER yields the strongest overall retention and highest AUC, confirming that replaying real trajectories remains the most effective strategy for mitigating forgetting. However, we treat ER as a reference rather than a directly comparable baseline, since it assumes access to stored demonstrations from previous tasks, which violates the constraints of our setting. Similarly, RAR, which replays simulator-rendered rollouts, performs close to ER, indicating that the remaining gap is attributable largely to generation fidelity rather than to the replay mechanism. While effective in simulation, it is impractical in real-world robotics, where replay generation requires re-deploying the robot and re-interacting with the environment to collect trajectories.

In contrast, REGEN substantially reduces forgetting using only pseudo-trajectories generated by the WAM itself, without storing any real demonstrations. Compared to Seq-FT, replaying pseudo-trajectories preserves forward transfer while significantly improving retention, reducing NBT by more than 50% across both suites. These results demonstrate that the generative capabilities of WAMs can serve as an effective form of implicit memory for continual robot learning.

REGEN in the Absence of Prior-Task Objects. REGEN initializes replay from an observed scene, implicitly assuming that it contains the objects required by prior tasks. This is a natural assumption for continual robot learning: for a robot to execute a previously learned manipulation task, the corresponding objects must be present in its environment. Nevertheless, we consider a

more restrictive stress-test using LIBERO-Spatial, where objects required by prior tasks may be absent from the current scene. In this case, the seed observation provides insufficient visual context for reconstructing the prior task. We therefore provide only the prior-task object metadata (object identities and locations) and use the simulator to render an initial seed observation. From this seed onward, REGEN generates the entire replay trajectory recurrently using the WAM, without access to stored demonstrations or trajectories. Even under this restrictive setting, REGEN substantially reduces forgetting while preserving forward transfer, recovering 86% of the AUC gap between Seq-FT and ER (Table. 2). Thus, when the visual prerequisites for a prior task are absent, lightweight scene metadata is sufficient to initialize REGEN without requiring stored demonstrations.

Table 2: Continual learning performance on LIBERO-Spatial where REGEN seeds replay from prior-task object configurations.*Requires stored demonstrations or environment interaction.

Method	FWT ↑	NBT ↓	AUC ↑
Seq-FT (Liu et al., 2023)	**87.4**	99.8	10.8
Seq-LoRA (Lee et al., 2024)	80.0	99.6	8.9
EWC (Kirkpatrick et al., 2017)	86.4	99.9	10.3
PackNet (Mallya & Lazebnik, 2018)	87.2	100	10.4
REGEN (ours)	87.2	**17.6**	**76.9**
ER* (Chaudhry et al., 2019)	86.4	-0.28	87.8
RAR* (ours)	87.0	-0.02	85.8

Table 3: Results of REGEN in real-world manipulation environment. *requires stored demonstrations from prior tasks, reported as an upper-bound reference. Bold denotes the best results among methods operating in our setting.

Method	FWT ↑	NBT ↓	AUC ↑
Seq-FT	74.4	94.6	31.9
REGEN (ours)	**91.1**	**53.9**	**60.4**
ER*	94.4	-1.7	95.9

5.3 REAL-WORLD EXPERIMENTS

Setting. We conduct all real-world experiments on an xArm7 robotic manipulator. Our setup is shown in Figure 3. We evaluate REGEN on three real-world manipulation tasks with a shared pick-and-place structure but different object-goal combinations: *(T1) Put carrot in bowl*, *(T2) Put carrot on plate*, and *(T3) Put eggplant in bowl*. Tasks are introduced sequentially in the order T1 → T2 → T3, with continual adaptation performed between stages. We collect 50 teleoperated demonstrations per task at a control frequency of 15 Hz. We report the continual learning metrics averaged across 3 seeds, with 10 randomized trials per seed varying in object placements and initial gripper configurations.

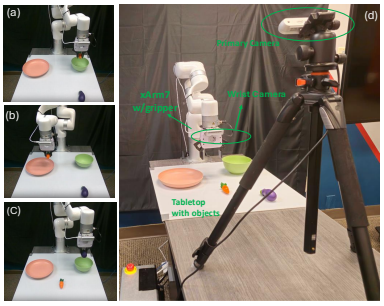


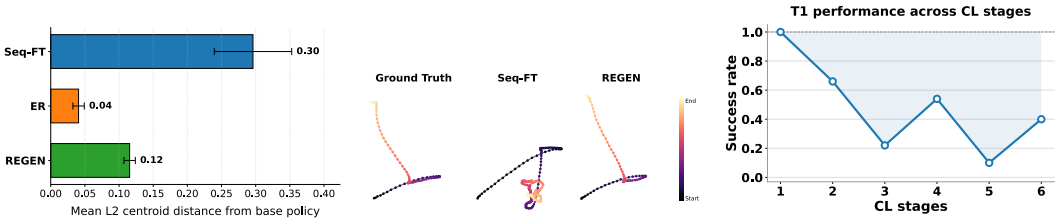
Figure 3: **Real-world setting.** a), b), c) illustrate tasks T1, T2, and T3, respectively. d) Our setup consists of an xArm7 manipulator, a wrist-mounted gripper camera, and a third-person RGB-D camera.

Results. Table 3 reports the continual learning performance on real-robot tasks. REGEN substantially outperforms Seq-FT, reducing NBT from 94.6 to 53.9 (approximately 43% less forgetting) while improving FWT from 74.4 to 91.1. We attribute the improved forward transfer to the regularizing effect of replayed pseudo-trajectories, particularly in the low-data regime where the base policy is initialized from a single task and therefore exhibits a limited prior over manipulation behaviors. These results demonstrate that REGEN extends effectively beyond simulation and provides a practical approach for continual learning on real robotic systems.

5.4 DIAGNOSTIC ANALYSIS OF REGEN

In this section, we analyze how REGEN preserves the policy's internal representations and behavior during continual learning, and then examine how far this preservation extends, over longer task sequences and under shifts in scene.

Action representation drift. Figure 4a measures the drift in action representations after the first continual learning stage. Across six base-stage tasks, we compute the centroid of the action latent representations under both the base policy and the continually adapted policy, and report the mean ℓ_2 distance between them. Seq-FT exhibits substantial representation drift (up to 0.3), consistent with its severe catastrophic forgetting. In contrast, both ER (0.04) and REGEN (0.12) maintain sig-



(a) (b) (c)

Figure 4: **(a)** Action representation drift from the base policy after the first continual learning stage for Seq-FT, ER, and REGEN. **(b)** XY-projection of trajectories predicted by Seq-FT and REGEN on a previously seen task, compared with the ground-truth demonstration. **(c)** REGEN under longer task sequences.

nificantly lower drift despite REGEN relying solely on generated pseudo-trajectories. These results suggest that REGEN effectively preserves the action representations learned by the base policy.

Visualization of predicted actions. In Figure. 4b, we compare rollouts of previous task after fine-tuning WAM using Seq-FT and REGEN against ground-truth demonstrations by projecting trajectories onto the XY plane. Seq-FT rollouts deviate substantially from the ground-truth trajectories, exhibiting erratic motion patterns indicative of catastrophic forgetting. In contrast, REGEN closely matches with the ground-truth trajectory in both shape and temporal progression, indicating that the policy retains the underlying structure of previously acquired manipulation skills.

Longer task horizon. We demonstrate the robustness of REGEN over longer task sequences by extending the continual sequence on LIBERO-Object from our standard four stages to six and report the success rate of the base task after every stage in Figure. 4c. Despite some degradation, REGEN retains the base task performance across all subsequent stages, and performance recovers after each drop, indicating that base-task retention under REGEN is robust to longer task horizon.

Does REGEN hold under scene covariate shift? We evaluate REGEN when the appearance of the scene changes between continual-learning stages. The base policy is pretrained in a wall-floor scene and sequentially adapted to a new task in a wood-floor scene of the LIBERO-Object suite (Figure 5). REGEN reduces NBT on the base-stage tasks from 83.3 to 20.5 compared to Seq-FT. Therefore, REGEN remains effective when the deployment scene differs in appearance from the base policy, which is important in the real-world, since a policy is rarely adapted in the scene its prior skills were learned.



Figure 5: Base stage (wall) and continual adaptation stage (wood) scenes.

5.5 ABLATION STUDY

We ablate the key design choices of REGEN on LIBERO-Object and LIBERO-Goal, to understand where forgetting occurs within the WAM, how many replays are needed, and pseudo-trajectory termination criteria.

Replay loss. We ablate the replay loss on the action and future-observation branches over a single continual stage on LIBERO-Object (Table 4). Supervising replays with action loss alone preserves most of it (NBT 46.6), while supervising the future-observation loss alone recovers far less (71.6), showing that forgetting occurs predominantly in the action branch. Supervising both the branches further reduces forgetting suggesting that future-observation supervision has a regularizing effect that further improves action retention.

Number of replays. We study the effect of the number of pseudo-replays from REGEN on continual learning performance. On LIBERO-Object (Tab. 5), we find that 10 replays per task achieve lower NBT than 5 replays while maintaining comparable FWT and AUC, indicating that additional replay diversity improves backward transfer.

Termination criterion. We compare the goal-reward criterion against rolling out to a fixed horizon $T_{\max}$. On LIBERO-Goal, reward-based termination yields higher-fidelity pseudo-trajectories (Tab. 6), since stopping at predicted task completion avoids extending the rollout into frames degraded by recursive prediction.

Table 4: Ablation on replay loss

$\mathcal{L}_{\text{act}}$	$\mathcal{L}_{\text{obs}}$	FWT ↑	NBT ↓	AUC ↑
✓	✗	95	46.6	57.7
✗	✓	95.2	71.6	36.0
✓	✓	**95.6**	**41.7**	**62.2**

Table 5: Effect of the number of pseudo-replays.

# No. of Replays	FWT ↑	NBT ↓	AUC ↑
5	96.5	33	62.5
10	96.1	31	62.3

Table 6: Effect of the termination criterion in REGEN.

Termination criterion	PSNR (↑)
Fixed Horizon with $T_{\max}$=200	18.4
Fixed Horizon with $T_{\max}$=150	19.5
Goal-reward function	20.3

6 LIMITATIONS OF REGEN & FUTURE WORK

Although REGEN substantially mitigates catastrophic forgetting without access to previous-task demonstrations, performance gap remains relative to the privileged ER baseline. We identify the primary bottleneck as the limited generative fidelity of current WAMs.

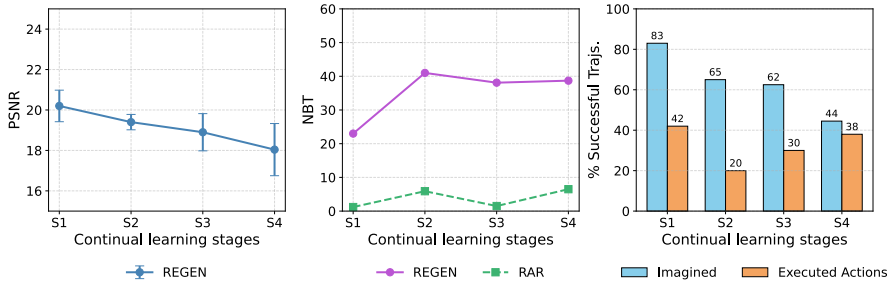


Figure 6: **Generative fidelity limits replay quality. (Left)** PSNR of REGEN-generated trajectories declines across continual stages on LIBERO-Goal. **(Middle)** NBT for REGEN and RAR; simulator-rendered observations recover most of the gap. **(Right)** Imagined success, from generated observations, against executed success, obtained by executing the predicted actions in the simulator.

Visual fidelity of generated observations. The first limitation concerns the quality of pseudo-trajectories generated across continual learning stages. Figure 6 (Left) shows a monotonic decline in the PSNR of synthesized trajectories as continual adaptation progresses, with qualitative examples shown in Appendix D.3. We attribute this to two compounding factors: visual artifacts and blurriness in generated observations that are recursively reused as conditioning inputs, and accumulation of error as the model is repeatedly updated across CL stages. Importantly, the reduction in PSNR correlates with increased NBT, suggesting that lower-fidelity pseudo-trajectories provide progressively weaker supervisory signals for retaining previously learned behaviors. The comparison with RAR isolates this effect: substituting simulator rollouts with WAM-generated pseudo-trajectories recovers most of the gap (Figure 6, middle), indicating that generative fidelity is a key limiting factor for replay quality.

Mismatch between predicted states and actions. A second limitation is the inconsistency between generated observations and predicted actions. In CL stage 1, 83% of rollouts appear successful visually while only 42% succeed on actual execution of actions in simulator (Figure 6, right). Thus, WAM imagines visually plausible successful trajectories while producing actions that are physically insufficient to complete the task. (Appendix D.4). We attribute this decoupling to degradation in the conditioning observations, which weakens the correspondence between visual prediction and action dynamics. Filtering generated replays can be explored in future work, though improving generative consistency and fidelity remains the principal direction for closing the gap to experience replay.

7 CONCLUSION

In this work, we present Recurrent Generative Replay (REGEN), the first continual learning framework to leverage the generative capabilities of WAMs as a native replay mechanism. By synthesizing pseudo-demonstrations conditioned on prior task instructions, current visual observations, and its own generated observations, REGEN mitigates catastrophic forgetting without requiring access to stored demonstrations from previous tasks. Experiments in both simulation and real-world manipulation demonstrate its effectiveness for continual robot learning. Finally, our analysis identifies long-horizon degradation in generated replays as the primary bottleneck, highlighting an important direction for future advances in WAMs and generative replay.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, including refining the language. We also used them for literature search assistance in identifying related work. We have not used generative AI tools for research ideation, experimental design, method development, code generation for experiments, data analysis, or the generation of results, figures, or claims. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Implementation details, hyperparameters, and additional qualitative results are provided in the appendix. Code, Data and Models will be released upon publication to facilitate further research.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025. URL https://arxiv.org/abs/2503.06669.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025a.

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, 2025b. URL https://arxiv.org/abs/2505.06111.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

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, 2024. URL https://arxiv.org/abs/2303.04137.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov,

Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Chongkai Gao, Haichuan Gao, Shangqi Guo, Tianren Zhang, and Feng Chen. Cril: Continual robot imitation learning via generative and prediction model. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6747–5754. IEEE, 2021.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution, https://worldmodels.github.io.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Daehee Lee, Minjong Yoo, Woo Kyung Kim, Wonje Choi, and Honguk Woo. Incremental learning of retrievable skills for efficient continual task adaptation. *Advances in Neural Information Processing Systems*, 37:17286–17312, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Huihan Liu, Changyeon Kim, Bo Liu, Minghuan Liu, and Yuke Zhu. Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning, 2026a. URL https://arxiv.org/abs/2603.03818.

Yuan Liu, Haoran Li, Shuai Tian, Yuxing Qin, Yuhui Chen, Yupeng Zheng, Yongzhen Huang, and Dongbin Zhao. Towards long-lived robots: Continual learning vla models via reinforcement fine-tuning, 2026b. URL https://arxiv.org/abs/2602.10503.

Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models, 2024. URL https://arxiv.org/abs/2310.05905.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning, 2018. URL https://arxiv.org/abs/1711.05769.

NVIDIA. Cosmos-predict2: World simulation model for physical ai, 2025. URL https://github.com/nvidia-cosmos/cosmos-predict2.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas, 2025. URL https://arxiv.org/abs/2512.15692.

Minting Pan, Wendong Zhang, Geng Chen, Xiangming Zhu, Siyu Gao, Yunbo Wang, and Xiaokang Yang. Continual visual reinforcement learning with a life-long world model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 146–162. Springer, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Kaushik Roy, Akila Dissanayake, Brendan Tidd, and Peyman Moghadam. M2distill: Multi-modal distillation for lifelong imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1429–1435, 2025. doi: 10.1109/ICRA55743.2025.11128857.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Ralf Römer, Yi Zhang, Yuming Li, and Angela P. Schoellig. Clare: Continual learning for vision-language-action models via autonomous adapter routing and expansion. *IEEE Robotics and Automation Letters*, pp. 1–8, 2026. ISSN 2377-3774. doi: 10.1109/lra.2026.3693992. URL http://dx.doi.org/10.1109/LRA.2026.3693992.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*, 2026.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay, 2017. URL https://arxiv.org/abs/1705.08690.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang, Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An advanced world action model for robot control, 2026. URL https://arxiv.org/abs/2604.27792.

Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, Yaping Li, Ping Wang, Junhao Cai, Jia Zeng, Hao Dong, and Jiangmiao Pang. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery, 2024. URL https://arxiv.org/abs/2311.02058.

Yuxuan Wu, Guangming Wang, Zhiheng Yang, Tianchen Deng, Maoqing Yao, Brian Sheil, and Hesheng Wang. Continually evolving skill knowledge in vision language action model, 2026. URL https://arxiv.org/abs/2511.18085.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust fine-tuning of vision-language-action robot policies via parameter merging. *arXiv Pre-print*, 2025. URL https://arxiv.org/abs/2512.08333.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world-action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026b. URL https://arxiv.org/abs/2602.15922.

Fanqi Yu, Matteo Tiezzi, Tommaso Apicella, Cigdem Beyan, and Vittorio Murino. Lifelong imitation learning with multimodal latent replay and incremental adjustment, 2026. URL https://arxiv.org/abs/2603.10929.

William Yue, Bo Liu, and Peter Stone. t-dgr: A trajectory-based deep generative replay method for continual learning in decision making. *arXiv preprint arXiv:2401.02576*, 2024.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Zangwei Zheng, Xiangyu Peng, Yuxuan Lou, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in $200k, 2026. URL https://arxiv.org/abs/2503.09642.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. URL https://arxiv.org/abs/2504.02792.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.