

World Action Models Enable Continual Imitation Learning with Generative Replays

Manish Kumar Govind, Dominick Reilly, Hieu Le, Srijan Das

Department of Computer Science
University of North Carolina at Charlotte United States
mgovind@charlotte.edu

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 by up to 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.

Keywords: Continual Imitation Learning, World Action Model, Generative Replay, Robot Control

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 [1, 2, 3, 4]. 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 [5, 6, 7]. 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 [8, 9] and more recent Vision-Language-Action (VLA) models [10, 11, 12]. 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 [13, 14, 15, 16, 17]. 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 [18, 19, 20], whereas the emerging paradigm is centered around pretrained robot foundation models trained on large-scale proprietary datasets that are rarely released publicly [11, 12, 1]. 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 [1, 2, 3, 4], 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 depicted in Figure 1.

Consequently, we propose **Recurrent Generative 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 [13] 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.

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 [5]. Existing approaches fall into three families: *regularization-based* methods that constrain weight updates [21, 22], *rehearsal-based* methods that retain real samples from prior tasks [23], and *architecture-based* methods that allocate task-specific parameters [24, 25]. Prior work [13] adapts these to lifelong robot manipulation, and recent methods extend them with skill discovery, distillation, or task-specific adapters combined with replay [15, 26, 27, 28]; a parallel line of work develops continual learning for VLAs using replay [17, 29, 16]. Across this line of work, forgetting is mitigated by storing real demonstrations, regularizing weights,

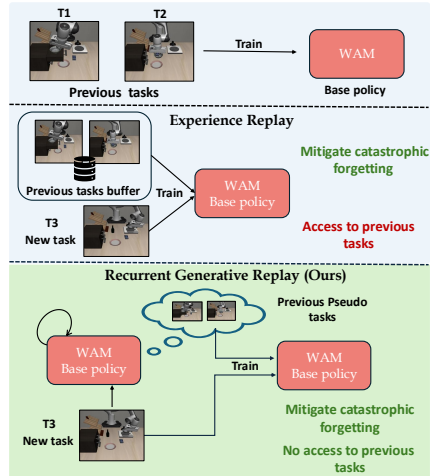


Figure 1: **Comparison of REGEN with Experience Replay.** Unlike standard Experience Replay (Middle), which requires privileged access to previous tasks, REGEN (Bottom) exploits generative property of WAMs to synthesize pseudo-demonstrations, mitigating forgetting without a physical replay buffer.

or partitioning parameters. 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 [30]. In robotics, CRIL [31], t-DGR [32], and [33] 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 [2] for both action and future-frame prediction, REGEN avoids model-based planning and eliminates the need for separate dynamics models.

World Models for robot control World models learn predictive representations of environment dynamics, originally proposed for sample-efficient reinforcement learning (RL) [34] and popularized by the Dreamer framework [35, 36]. While traditionally used for RL, the recent scaling of internet-level video data has enabled a new class of world-action models (WAMs) for imitation learning [2, 3, 37, 1, 38, 39]. WAMs leverage unified architectures to jointly predict actions and future observations. While these models internalize physical causality, adapting such high-capacity architectures to non-stationary task streams remains an open continual learning challenge.

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 $\mathbf{I}_t^1, \dots, \mathbf{I}_t^n$ 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 [40, 41, 42, 43], 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

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

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 $\mathcal{T}_{\text{prev}} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_M\}$. 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 into 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, because this objective contains no explicit information about previous tasks, it can lead to catastrophic forgetting. Replay-based continual learning methods mitigate this issue by jointly training on stored demonstrations from prior tasks and current-task data. Although such approaches are infeasible in our setting due to the absence of previous trajectories, WAMs provide a unique advantage in this regime. Since WAMs jointly model future actions and future visual observations, they expose a generative interface capable of simulating prior task dynamics conditioned only on task instructions. We leverage this property to enable continual robot learning without storing or replaying ground-truth demonstrations from previous tasks.

4.2 Recurrent Generative Replay

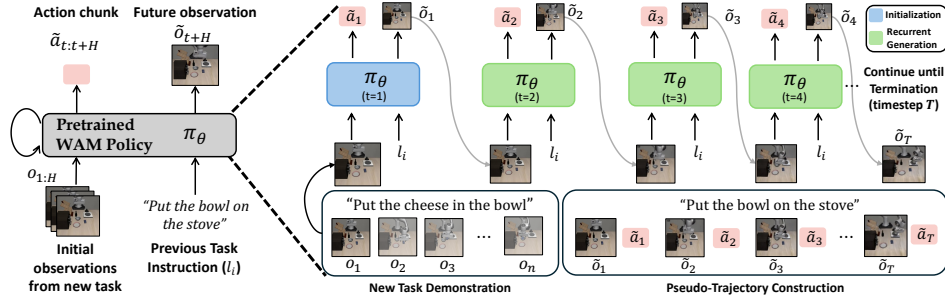


Figure 2: **Overview of the pseudo-trajectory generation process.** *Left:* unrolled view of the REGEN. *Right:* REGEN rolls out π_θ recurrently to construct a pseudo-trajectory for a previous task. The policy is seeded with new-task observations and the previous task’s instruction ℓ_i (initialization, 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 ($H \leq t \leq T_{\text{max}}$). 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:

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

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:

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

where $\tilde{\mathbf{a}}_t$ denotes the first action of the chunk predicted at time step t , and $T_i \leq T_{\max}$ 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 $\mathcal{D}_k^+ = \mathcal{D}_k \cup \mathcal{R}_k$.

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:

$$\min_{\theta} \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, \ell) \sim \mathcal{D}_k^+} [\mathcal{L}_{\text{BC}}(\pi_{\theta}(\mathbf{o}_t, \ell), \mathbf{a}_t)], \quad (4)$$

where \mathcal{L}_{BC} denotes the behavioral cloning loss. 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 Experimental Results

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 [2] as the underlying WAM, initialized from Cosmos-Predict2-2B [40]. 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 fine-tunes the policy for 2K iterations from the checkpoint of the previous stage. For replay generation, we synthesize 10 pseudo-trajectories per previous task in $\mathcal{T}_{\text{prev}}$. Additional architectural details, hyperparameters and ablations are provided in the Appendix.

For evaluation, we report three standard continual learning metrics: Forward Transfer (FWT) [13, 15], Negative Backward Transfer (NBT) [29], and Area Under the Curve (AUC) [13, 15]. All metrics are computed using task success rates. Let N denote the total number of tasks in the continual learning sequence, and let $r_{i,j}$ denote the success rate on task j after training up to task i . Then, FWT measures the ability to acquire new tasks, while NBT quantifies forgetting on previously learned tasks:

$$\mathbf{FWT} = \frac{1}{N} \sum_{n=1}^N r_{n,n}, \quad \mathbf{NBT} = \frac{1}{N-1} \sum_{n=1}^{N-1} \mathbf{NBT}_n, \quad \mathbf{NBT}_n = \frac{1}{N-n} \sum_{p=n+1}^N \left(\frac{r_{n,n} - r_{p,n}}{r_{n,n}} \right)$$

Higher FWT indicates stronger forward transfer, whereas lower NBT corresponds to reduced forgetting. Finally, AUC measures overall performance across both current and previously learned tasks:

$$\mathbf{AUC} = \frac{1}{N} \sum_{n=1}^N \frac{1}{N-n+1} \left(r_{n,n} + \sum_{p=n+1}^N r_{p,n} \right).$$

Higher AUC indicates better overall continual learning performance.

5.2 LIBERO Simulated Environment

Setting. We conduct our simulation experiments on LIBERO benchmark [13]. 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. Our continual learning setup involves two stages, similar to [15, 44] (1) a base stage and (2) a continual learning stage. In the base stage, six tasks are used for pretraining. During continual learning, the remaining four tasks are introduced sequentially, with one new task added at each continual learning stage. We follow the task ordering defined for each benchmark suite in [29].

Baseline Methods. We evaluate REGEN against three methods. i) **Sequential Fine-Tuning (Seq-FT)** [13] is the naive baseline, fine-tuning the base policy on each new task without any mechanism to preserve prior skills. ii) **Experience Replay (ER)** where we assume access to real demonstrations from prior tasks and mixes them into training. This setting adopted from [13, 15] is different from ours, thus we include it as an upper-bound reference. iii) **Rollouts-as-Replay (RAR)** as an alternative to noisy observations generated by REGEN (Sec. 6), like REGEN, stores no real trajectories: at each stage, we rollout previous-task policies in the simulator and use the resulting rollouts as replay data, to compare *simulator-rendered* and *model-generated* replay.

Table 1: Comparison of REGEN with traditional continual learning methods on LIBERO benchmarks.

Method	LIBERO-Object			LIBERO-Goal			Method	LIBERO-Spatial		
	FWT \uparrow	NBT \downarrow	AUC \uparrow	FWT \uparrow	NBT \downarrow	AUC \uparrow		FWT \uparrow	NBT \downarrow	AUC \uparrow
Seq-FT [13]	92.7	82.6	24.9	90.6	100	10.3	Seq-FT [13]	87.4	99.8	10.8
ER [23]	95.7	4.8	93.4	94.0	7.2	92.4	ER [23]	86.4	-0.28	87.8
RAR	96.9	3.0	95.2	92.8	15.4	82.6	RAR	87	-0.02	85.8
REGEN (ours)	95.3	26.1	65.5	90.6	44.9	40.8	REGEN \dagger (ours)	87.2	17.6	76.9

Results. Table 1 reports continual learning performance on the three LIBERO benchmark suites. 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 [13, 15, 28]. Using 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. RAR mitigates forgetting by replaying simulator-collected rollouts, achieving performance close to ER with minimal forgetting and without storing real replay buffers. While effective in simulation, such approaches are 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 sequential fine-tuning, replaying pseudo-trajectories preserves forward transfer while significantly improving retention, reducing NBT by more than 50% across multiple suites. These results demonstrate that the generative capabilities of WAMs can serve as an effective form of implicit memory for continual robot learning. For LIBERO-Spatial, we introduce a variant denoted as **REGEN \dagger** , where replay generation is initialized using object configurations sampled from previous tasks. This modification is necessary because the benchmark evaluates spatial generalization across object arrangements, and replay generation requires the corresponding objects to be present in the scene. For example, trajectories involving an object absent from the current environment cannot be reliably synthesized.

VLA vs. WAM Continual Learning. Compared to pretrained VLAs such as $\pi_{0.5}$ [12], forgetting in WAMs is initially more severe; however, Table 2 shows that large-scale pretraining alone does not eliminate catastrophic forgetting. Despite being trained on substantially larger robotic datasets [18], $\pi_{0.5}$ still exhibits significant degradation on previously learned tasks during continual adaptation. In contrast, REGEN achieves stronger retention using only generated pseudo-replays, by leveraging the joint action-observation generative modeling capability unique to WAMs. Unlike standard VLAs [10, 12, 45], WAMs can explicitly generate future observations, enabling replay without

Table 2: VLA vs WAM: continual learning performance on LIBERO-Goal.

Policy	Method	FWT \uparrow	NBT \downarrow	AUC \uparrow
$\pi_{0.5}$ [12]	Seq-FT	96.8	88	35.5
	Seq-FT	90.6	100	10.3
Cosmos-Policy [2]	REGEN (ours)	<u>90.6</u>	38.7	40.8

Table 3: Results of REGEN in real-world manipulation environment.

Method	FWT \uparrow	NBT \downarrow	AUC \uparrow
Seq-FT	50	96.3	13.8
REGEN (ours)	80	60.5	53.8

additional data collection or storage. These results suggest that WAMs provide a more suitable foundation for lifelong robot learning.

5.3 Real-world Single-arm Manipulation

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*: place the carrot inside the bowl; (T2) *Put carrot on plate*: place the carrot on the plate; and (T3) *Put eggplant in bowl*: place the eggplant inside the bowl. Tasks are introduced sequentially in the order T1 \rightarrow T2 \rightarrow T3, with continual adaptation performed between stages. We collect 50 teleoperated demonstrations per task at a control frequency of 15 Hz. Evaluation metrics are averaged across the two continual learning stages, and each policy is evaluated over 10 randomized trials with varying object placements and initial gripper configurations.

Results. Table 3 reports the real-world continual learning results. REGEN substantially outperforms sequential fine-tuning, reducing NBT from 96.3 to 60.5 (approximately 40% less forgetting) while improving FWT from 50 to 80. 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.

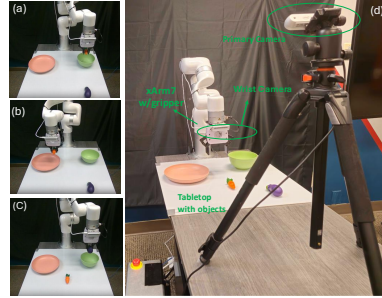


Figure 3: **Real-world setting.** a), b), c) illustrate tasks T1, T2, and T3, respectively. d) Our setup consists of an xArm7 manipulator, a wrist-mounted gripper camera, and a third-person RGB-D camera.

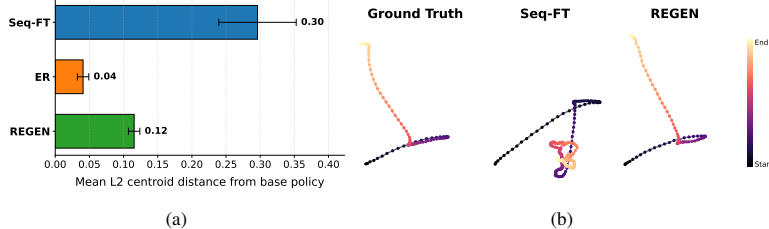


Figure 4: (a) Action representation drift from the base policy after the first continual learning stage between Seq-FT, ER, and REGEN (b) XY-projection of trajectories predicted by Seq-FT and REGEN on a previously seen task, compared with the ground-truth demonstration.

5.4 Representation Analyses

We further analyze how REGEN preserves the policy’s internal representations and behaviors throughout continual learning.

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 significantly 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 Fig. 4b, we additionally compare action trajectories generated by Seq-FT and REGEN against ground-truth demonstrations on a previously learned evaluation task by projecting trajectories onto the XY plane. Seq-FT produces trajectories that deviate substantially from the ground-truth behavior, exhibiting erratic motion patterns indicative of catastrophic forgetting. In contrast, trajectories generated by REGEN closely match the ground-truth trajectory in both shape and temporal progression, indicating that the policy retains the underlying structure of previously acquired manipulation skills.

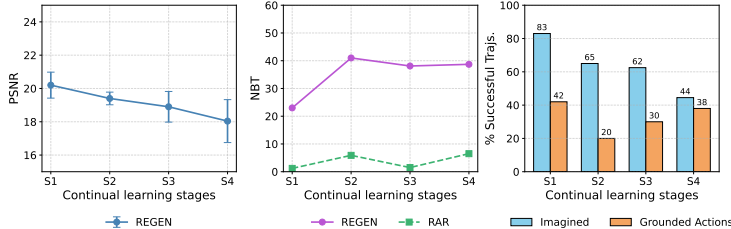


Figure 5: (Left) PSNR (with std) of generated trajectories from REGEN across CL stages on LIBERO-Goal. (Middle) NBT comparison between REGEN and RAR. (Right) successful imagined trajectories vs. successful action grounded trajectories on LIBERO-Goal.

6 Limitations of REGEN & Future Direction

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

Visual fidelity of generated observations. The first limitation concerns the quality of pseudo-trajectories generated across successive continual learning stages. Figure 5 (Left) shows a monotonic decline in the PSNR of synthesized trajectories as continual adaptation progresses. We attribute this degradation to two compounding factors: (i) visual artifacts and blurriness in generated observations that are recursively reused as conditioning inputs for future prediction, and (ii) accumulation of errors as the model is repeatedly updated over continual learning 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. This observation is further supported by the comparison with RAR. Replacing simulator-generated replay trajectories with WAM-generated pseudo-trajectories leads to a substantial drop in continual learning performance (Fig. 5, middle), indicating that generative fidelity is a key limiting factor for replay quality.

Mismatch between predicted states and actions. A second limitation arises from inconsistencies between generated future observations and the corresponding predicted actions. Figure 5 (Right) compares the *imagined* success rate, evaluated from generated observations, with the *grounded* success rate obtained by executing the predicted actions in the simulator. At Stage 1, the model achieves an imagined success rate of 83% but only a grounded success rate of 42%. This discrepancy indicates that the WAM frequently predicts visually plausible successful outcomes while generating actions that are physically insufficient to accomplish the task. We hypothesize that this decoupling stems from degradation in generated observations, which weakens the alignment between visual predictions and action dynamics. Consequently, improving the generative consistency and visual fidelity of WAMs remains a future direction for narrowing the performance gap to experience replay.

7 Conclusion

We presented 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 observations as the primary bottleneck, highlighting an important direction for future advances in WAMs and generative replay.

Acknowledgments

This work was supported in part by the National Science Foundation (IIS-2245652) and the University of North Carolina at Charlotte. Computational resources were provided by the NSF National AI Research Resource Pilot (NAIRR240338) and NCShare.

References

- [1] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, A. Malik, K. Lee, W. Liang, N. Ranawaka, J. Gu, Y. Xu, G. Wang, F. Hu, A. Narayan, J. Bjorck, J. Wang, G. Kim, D. Niu, R. Zheng, Y. Xie, J. Wu, Q. Wang, R. Julian, D. Xu, Y. Du, Y. Chebotar, S. Reed, J. Kautz, Y. Zhu, L. J. Fan, and J. Jang. World action models are zero-shot policies, 2026. URL <https://arxiv.org/abs/2602.15922>.
- [2] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.
- [3] 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.
- [4] A. Ye, B. Wang, C. Ni, G. Huang, G. Zhao, H. Li, H. Li, J. Li, J. Lv, J. Liu, M. Cao, P. Li, Q. Deng, W. Mei, X. Wang, X. Chen, X. Zhou, Y. Wang, Y. Chang, Y. Li, Y. Zhou, Y. Ye, Z. Liu, and Z. Zhu. Gigaworld-policy: An efficient action-centered world-action model. *arXiv preprint arXiv:2603.17240*, 2026.
- [5] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [6] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. 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.
- [7] I. Shenfeld, J. Pari, and P. Agrawal. RL’s razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*, 2026.
- [8] 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, 2024. URL <https://arxiv.org/abs/2303.04137>.
- [9] T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [10] 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, 2024. URL <https://arxiv.org/abs/2406.09246>.
- [11] 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, 2024. URL <https://arxiv.org/abs/2410.24164>.
- [12] P. 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, 2025. URL <https://arxiv.org/abs/2504.16054>.
- [13] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.
- [14] Y. Zhu, P. Stone, and Y. Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.
- [15] W. Wan, Y. Zhu, R. Shah, and Y. Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery, 2024. URL <https://arxiv.org/abs/2311.02058>.
- [16] Y. Liu, H. Li, S. Tian, Y. Qin, Y. Chen, Y. Zheng, Y. Huang, and D. Zhao. Towards long-lived robots: Continual learning via models via reinforcement fine-tuning, 2026. URL <https://arxiv.org/abs/2602.10503>.
- [17] Y. Wu, G. Wang, Z. Yang, T. Deng, M. Yao, B. Sheil, and H. Wang. Continually evolving skill knowledge in vision language action model, 2026. URL <https://arxiv.org/abs/2511.18085>.
- [18] O. X.-E. Collaboration, A. O’Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irgan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, G. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, M. Z. Irshad, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart’in-Mart’in, R. Bajjal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, V. Guizilini, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu,

- Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023.
- [19] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, X. He, X. Huang, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.
- [20] Y. Tian, Y. Yang, Y. Xie, Z. Cai, X. Shi, N. Gao, H. Liu, X. Jiang, Z. Qiu, F. Yuan, Y. Li, P. Wang, J. Cai, J. Zeng, H. Dong, and J. Pang. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025.
- [21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [22] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/zenke17a.html>.
- [23] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.
- [24] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning, 2018. URL <https://arxiv.org/abs/1711.05769>.
- [25] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [26] K. Roy, A. Dissanayake, B. Tidd, and P. Moghadam. M2distill: Multi-modal distillation for lifelong imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1429–1435, 2025. doi:10.1109/ICRA55743.2025.11128857.
- [27] Z. Liu, J. Zhang, K. Asadi, Y. Liu, D. Zhao, S. Sabach, and R. Fakoore. Tail: Task-specific adapters for imitation learning with large pretrained models, 2024. URL <https://arxiv.org/abs/2310.05905>.
- [28] R. Römer, Y. Zhang, Y. Li, and A. P. Schoellig. Clare: Continual learning for vision-language-action models via autonomous adapter routing and expansion. *IEEE Robotics and Automation Letters*, page 1–8, 2026. ISSN 2377-3774. doi:10.1109/lra.2026.3693992. URL <http://dx.doi.org/10.1109/LRA.2026.3693992>.
- [29] H. Liu, C. Kim, B. Liu, M. Liu, and Y. Zhu. Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning, 2026. URL <https://arxiv.org/abs/2603.03818>.
- [30] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay, 2017. URL <https://arxiv.org/abs/1705.08690>.
- [31] C. Gao, H. Gao, S. Guo, T. Zhang, and F. Chen. Cril: Continual robot imitation learning via generative and prediction model. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6747–5754. IEEE, 2021.

- [32] W. Yue, B. Liu, and P. Stone. t-dgr: A trajectory-based deep generative replay method for continual learning in decision making. *arXiv preprint arXiv:2401.02576*, 2024.
- [33] M. Pan, W. Zhang, G. Chen, X. Zhu, S. Gao, Y. Wang, and X. Yang. Continual visual reinforcement learning with a life-long world model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 146–162. Springer, 2025.
- [34] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pages 2451–2463. Curran Associates, Inc., 2018. URL <https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution>. <https://worldmodels.github.io>.
- [35] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL <https://arxiv.org/abs/1912.01603>.
- [36] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models, 2024. URL <https://arxiv.org/abs/2301.04104>.
- [37] AgiBot-World-Contributors, Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, S. Jiang, Y. Jiang, C. Jing, H. Li, J. Li, C. Liu, Y. Liu, Y. Lu, J. Luo, P. Luo, Y. Mu, Y. Niu, Y. Pan, J. Pang, Y. Qiao, G. Ren, C. Ruan, J. Shan, Y. Shen, C. Shi, M. Shi, M. Shi, C. Sima, J. Song, H. Wang, W. Wang, D. Wei, C. Xie, G. Xu, J. Yan, C. Yang, L. Yang, S. Yang, M. Yao, J. Zeng, C. Zhang, Q. Zhang, B. Zhao, C. Zhao, J. Zhao, and J. Zhu. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025. URL <https://arxiv.org/abs/2503.06669>.
- [38] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. URL <https://arxiv.org/abs/2504.02792>.
- [39] M. Team, C. Xiang, F. Bao, H. Liu, H. Tan, H. Bi, J. Li, J. Liu, J. Pang, K. Jing, L. Liu, M. Cai, R. Cui, R. Zhao, R. Wang, S. Huang, Y. Feng, Y. Rong, Z. Wang, and J. Zhu. Motubrain: An advanced world action model for robot control, 2026. URL <https://arxiv.org/abs/2604.27792>.
- [40] NVIDIA. Cosmos-predict2: World simulation model for physical ai, 2025. URL <https://github.com/nvidia-cosmos/cosmos-predict2>.
- [41] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [42] T. Seedance, D. Chen, L. Chen, X. Chen, Y. Chen, Z. Chen, Z. Chen, F. Cheng, T. Cheng, Y. Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.
- [43] Z. Zheng, X. Peng, Y. Lou, C. Shen, T. Young, X. Guo, B. Wang, H. Xu, H. Liu, M. Jiang, W. Li, Y. Wang, A. Ye, G. Ren, Q. Ma, W. Liang, X. Lian, X. Wu, Y. Zhong, Z. Li, C. Gong, G. Lei, L. Cheng, L. Zhang, M. Li, R. Zhang, S. Hu, S. Huang, X. Wang, Y. Zhao, Y. Wang, Z. Wei, and Y. You. Open-sora 2.0: Training a commercial-level video generation model in \$200k, 2026. URL <https://arxiv.org/abs/2503.09642>.
- [44] F. Yu, M. Tiezzi, T. Apicella, C. Beyan, and V. Murino. Lifelong imitation learning with multimodal latent replay and incremental adjustment, 2026. URL <https://arxiv.org/abs/2603.10929>.
- [45] 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, 2025. URL <https://arxiv.org/abs/2505.06111>.