

# MTIL: Encoding Full History With Mamba for Temporal Imitation Learning

Yulin Zhou<sup>1</sup>, Yuankai Lin<sup>1</sup>, Fanzhe Peng<sup>1</sup>, Jiahui Chen<sup>1</sup>, *Member, IEEE*, Kaiji Huang,  
Hua Yang<sup>1</sup>, *Member, IEEE*, and Zhouping Yin<sup>1</sup>, *Member, IEEE*

**Abstract**—Standard imitation learning (IL) methods have achieved considerable success in robotics, yet often rely on the Markov assumption, which falters in long-horizon tasks where history is crucial for resolving perceptual ambiguity. This limitation stems not only from a conceptual gap but also from a fundamental computational barrier: prevailing architectures like Transformers are often constrained by quadratic complexity, rendering the processing of long, high-dimensional observation sequences infeasible. To overcome this dual challenge, we introduce Mamba Temporal Imitation Learning (MTIL). Our approach represents a new paradigm for robotic learning, which we frame as a practical synthesis of World Model and Dynamical System concepts. By leveraging the linear-time recurrent dynamics of State Space Models (SSMs), MTIL learns an implicit, action-oriented world model that efficiently encodes the *entire* trajectory history into a compressed, evolving state. This allows the policy to be conditioned on a comprehensive temporal context, transcending the confines of Markovian approaches. Through extensive experiments on simulated benchmarks (ACT, Robomimic, LIBERO) and on challenging real-world tasks, MTIL demonstrates superior performance against SOTA methods like ACT and Diffusion Policy, particularly in resolving long-term temporal ambiguities. Our findings not only affirm the necessity of full temporal context but also validate MTIL as a powerful and a computationally feasible approach for learning long-horizon, non-Markovian behaviors from high-dimensional observations.

**Index Terms**—Imitation learning, deep learning for grasping and manipulation, learning from demonstration.

## I. INTRODUCTION

IMITATION Learning (IL) has emerged as a powerful paradigm for teaching robots complex skills directly from expert demonstrations, bypassing the need for intricate reward engineering often required in reinforcement learning [1], [2], [3]. Behavioral Cloning (BC), the simplest form of IL, learns

a direct mapping from observations to actions via supervised learning and has enabled robots to perform a variety of tasks [1], [2], [3], [4]. Recent advancements, particularly leveraging powerful sequence models and generative approaches, have led to state-of-the-art (SOTA) methods such as the Action Chunking Transformer (ACT) [5], [6] and Diffusion Policy [7], [8], [9], [10], which excel at learning visuomotor control policies for complex manipulation. Despite these successes, a fundamental limitation persists in many current IL approaches: the reliance on the Markov assumption. These methods typically predict the action  $a_t$  based solely on the current observation  $o_t$  or a short, fixed-length history window  $o_{t-k:t}$  [1], [11], [12], [13]. This assumption breaks down in tasks where the history beyond this limited window is necessary to resolve ambiguity in the current observation. Consider a sequential task requiring a robot to first place an object at location A, and subsequently move it to location B. At an intermediate configuration, the robot's visual observation and proprioceptive state ( $o_t$ ) might be identical regardless of whether it has successfully completed the sub-task at location A. A Markovian policy, lacking the memory of visiting A, cannot distinguish these fundamentally different historical contexts and may erroneously proceed directly to B, failing the task [1], [14], [15]. This temporal ambiguity signifies an underlying Partially Observable Markov Decision Process (POMDP), posing a critical challenge for standard IL methods in state-dependent tasks.

Since ambiguous tasks manifest as POMDPs and human demonstrations are inherently non-Markovian, effective imitation necessitates history-aware policies. While the critical role of historical context has been increasingly recognized [12], a fundamental barrier has remained: the computational infeasibility of processing long, high-dimensional observation histories with prevailing architectures like Transformers. Addressing this, we introduce Mamba Temporal Imitation Learning (MTIL). Our approach is specifically designed to incorporate the complete observational history into decision-making by leveraging the unique properties of State Space Models (SSMs), particularly the recently developed Mamba architecture [16], [17]. Mamba's recurrent structure allows it to maintain a compressed hidden state  $h_t$  that theoretically encapsulates information from the entire preceding observation sequence  $H_t = (o_1, \dots, o_t)$ . Instead of relying solely on  $o_t$ , MTIL learns a policy  $\pi(a_t|h_t, o_t)$  that explicitly conditions the action prediction on this history-infused hidden state  $h_t$  in conjunction with the current observation  $o_t$ , enabling differentiation between observationally similar states

Received 18 May 2025; accepted 20 September 2025. Date of publication 29 September 2025; date of current version 10 October 2025. This article was recommended for publication by Associate Editor S. Bansal and Editor A. Faust upon evaluation of the reviewer's comments. This work was supported in part by the Joint Funds of the National Natural Science Foundation of China under Grant U222A20208, in part by the Natural Science Foundation Innovation Group Project of Hubei Province under Grant 2022CFA018, and in part by the Key Research and Development Program of Guangdong Province under Grant 2022B0202010001-2. (*Corresponding author: Hua Yang.*)

The authors are with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yulinzhou@hust.edu.cn; huayang@hust.edu.cn).

Project code are available at <https://github.com/yulinzhouZYL/MTIL>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3615520>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3615520

and the correct execution of complex sequential tasks. Our contributions are threefold:

- 1) We propose MTIL, a novel architecture that is the first to leverage the linear-time recurrence of State Space Models to make full-trajectory imitation learning from high-dimensional visual data computationally feasible on commodity hardware, breaking the quadratic bottleneck of attention-based models.
- 2) We provide a new theoretical framing for this approach, positing MTIL as learning an *implicit dynamical system*. This system's evolving state acts as a continuous belief-state representation, offering a robust solution to the core problem of temporal ambiguity in partially observable environments.
- 3) We provide extensive empirical validation demonstrating that MTIL significantly outperforms state-of-the-art methods, including ACT and Diffusion Policy. Furthermore, on tasks requiring long-term memory, MTIL also surpasses other full-history-capable baselines like Transformer-XL, validating the unique advantages of its underlying architecture.

## II. RELATED WORKS

### A. Markovian and Short-History Imitation Learning

A cornerstone of imitation learning, Behavioral Cloning (BC), typically learns a Markovian policy  $\pi(a_t|o_t)$  via supervised learning [1], [2], [3], [4]. Although fundamental, this approach inherently struggles with covariate shift and tasks that require memory beyond current observation [3], [11]. Many contemporary methods, despite advances, effectively operate within similar constraints or rely on limited observation histories. For instance, the Action Chunking Transformer (ACT) [5], [6], leveraging the Transformer architecture [18], predicts chunks of actions  $a_{t:t+K-1}$  conditioned on present observation and potentially a latent variable from a CVAE. Although action chunking improves temporal smoothness and reduces the effective horizon [19], [20], its temporal modeling is largely confined to short-term dependencies implicitly captured through the time aggregation of chunks while reasoning, potentially failing when resolving ambiguities requires longer context [5]. Similarly, Diffusion Policy [7], [8], [9], [10], while adept at capturing complex, multimodal action distributions [7], [21], commonly conditions the diffusion process on the present observation or a short history, limiting its capacity for tasks requiring long-term memory [7], [21]. While extensions like Diff-Control [21] introduce forms of statefulness, they differ fundamentally from MTIL's direct use of a recurrent SSM state to encode the full task history. Other techniques, including Implicit BC [22] and Energy-Based Models [23], also often operate primarily on the current state.

### B. Temporal and Sequential Imitation Learning

The inadequacy of the strict Markov assumption has long motivated efforts to incorporate temporal context. Early explorations employed Recurrent Neural Networks (RNNs) like LSTMs [1], [4], [12], [24]. However, these architectures face

challenges with long-term dependencies due to vanishing gradients [24]. Furthermore, practical implementations often resorted to fixed history windows and periodic state resets (e.g., sequence lengths of 10-50 in Robomimic [12], [25]), precluding the capture of full trajectory history. More recently, Transformer-based models (e.g., BeT [26], RT-1 [27], [28], OPTIMUS [29], ICRT [30], Baku [27], [28], MDT [31]) have become prominent, utilizing self-attention to model sequence correlations. However, the  $O(L^2)$  computational complexity of attention imposes practical limits on the size of the context window [1], [24], hindering their ability to efficiently process entire long trajectories. Even recurrent variants like Transformer-XL [32], while theoretically capable of processing long sequences, still rely on the computationally intensive attention mechanism. Distinct strategies for managing long horizons involve temporal abstraction. Hierarchical Imitation Learning [14], [15], [33], [34], [35] and Skill Chaining [14], [36], [37] decompose tasks, learning policies over skills or sub-goals. Waypoint-based methods like AWE [19] or primitive-based approaches like PRIME [15] operate at higher levels of abstraction. While effective, these approaches fundamentally differ from MTIL, which aims to directly model the complete low-level observation-action sequence history, potentially offering robustness against issues like error propagation in skill chaining [14], [37].

### C. State Space Models (SSMs) and Mamba in Robotics

State Space Models (SSMs) represent a compelling paradigm for sequence modeling, defined by their recurrent hidden state dynamics [16], [17], [38], [39], [40], [41], [42]. Mamba [16] marked a significant advancement, introducing input-dependent parameters ( $\mathbf{A}, \mathbf{B}, \mathbf{C}, \Delta$ ) via a selective scan mechanism. This allows Mamba models to dynamically focus on relevant sequence information while maintaining the linear time complexity characteristic of SSMs, synergizes the capacity for long-range dependency modeling, akin to Transformers, with the efficient recurrent updates reminiscent of RNNs, yet sidesteps the quadratic scaling bottlenecks of the former [1], [24] and the gradient propagation issues of the latter [24], achieving strong empirical results [16]. The robotics community has begun investigating Mamba's potential [16], [17], [43], [44]. For instance, MaIL [17] employed Mamba as an imitation learning backbone, showing promise particularly in low-data regimes [16]. Mamba Policy [45], [46] integrated Mamba structures within diffusion models to enhance efficiency, while X-IL [44] explored Mamba within a modular IL framework. While these works adeptly leverage Mamba's sequence processing power, MTIL distinguishes itself through its core premise: harnessing the step-updated recurrent state  $h_t$  as an explicit, dynamically built representation of the *entire* observation history. This approach, tightly coupled with its sequential training methodology, directly overcomes the temporal ambiguity challenges inherent in Markovian assumptions common in imitation learning.

## III. MAMBA TEMPORAL IMITATION LEARNING (MTIL)

We introduce Mamba Temporal Imitation Learning (MTIL), a novel imitation learning framework designed to overcome the

limitations of the Markov assumption by leveraging the full history of observations encoded within the recurrent state of an advanced State Space Model (SSM) architecture.

### A. Background and Motivation

Standard imitation learning often assumes a Markov Decision Process (MDP), learning reactive policies  $\pi(a_t|o_t)$  via Behavioral Cloning. However, observational ambiguity fundamentally renders many sequential tasks as Partially Observable MDPs (POMDPs) [1], where the optimal policy necessitates conditioning on the full history  $H_t = (o_1, a_1, \dots, o_t)$ . Theoretically, this history is captured by the belief state  $b_t = P(s_t|H_t)$ , dictating the optimal policy  $\pi^*(a_t|b_t)$  [47].

Directly computing or representing the belief state  $b_t$  is generally intractable. This motivates learning a compressed history representation  $h_t \approx b_t$  using recurrent models. This aligns with the core ideas of both World Models, which learn a predictive latent state of the world, and Dynamical Systems (DS) approaches to control, which rely on an evolving internal state. State Space Models (SSMs) like Mamba [16] offer a particularly compelling synthesis of these ideas. They provide a structured recurrent update  $h_t = f(h_{t-1}, x_t)$  (where  $x_t$  encodes  $o_t$ ) with linear time complexity  $O(L)$ . This efficiency is the critical enabler for tractably encoding the long sequences required for full history representation, overcoming the computational barriers of prior architectures [48].

Our motivation for MTIL stems from leveraging Mamba's state  $h_t$  as this potent, efficiently computed representation of the full history. We view  $h_t$  as the state of a learned, implicit dynamical system that acts as an action-oriented world model. By conditioning actions on both the current observation  $o_t$  and this history-infused state  $h_t$ , MTIL learns a non-Markovian policy:

$$\hat{a}_{t:t+K-1} \approx \pi(o_t, h_t)$$

thereby directly addressing the core challenge of decision-making under ambiguity in POMDP-structured imitation learning by effectively utilizing the entire history.

### B. Leveraging Full Trajectory History With Mamba-2

MTIL employs Mamba-2 [49], an advanced structured State Space Model (SSM) notable for its refined selective mechanism and efficiency [16]. Improving upon Mamba, Mamba-2 enhances hardware utilization and clarifies theoretical links to attention while retaining dynamic context adaptation via input-dependent parameters [49]. Its core lies in the discretized SSM recurrence governing the hidden state  $h_t \in \mathbb{R}^N$  evolution based on input  $x_t$  (derived from observation  $o_t$ ):

$$h_t = \bar{\mathbf{A}}_t h_{t-1} + \bar{\mathbf{B}}_t x_t$$

$$y_t = \mathbf{C}_t h_t$$

Crucially, the input-dependent parameters  $(\Delta_t, \bar{\mathbf{B}}_t, \mathbf{C}_t) = f(x_t)$  enable selective state dynamics. This allows the model to learn a highly non-linear dynamical system where the system matrices themselves adapt based on the current input. This

---

#### Algorithm 1: MTIL Training (Sequential Step-based).

---

**Require:** Expert trajectories  $\mathcal{D} = \{\tau_i\}$ ,  
 $\tau_i = (o_1, a_1, \dots, o_{T_i}, a_{T_i})$ , MTIL Policy  $\pi_\theta$ , initialized parameters  $\theta$ , Loss  $\mathcal{L}$  (MSE), Chunk size  $k$ , Optimizer  $Opt$

- 1: Initialize policy parameters  $\theta$
- 2: **for** each training epoch **do**
- 3:   **for** each trajectory  $\tau_i \in \mathcal{D}$  **do**
- 4:     Initialize hidden state  $h_0$ , trajectory loss  $\mathcal{L}_{traj} \leftarrow 0$
- 5:     **for**  $t = 0$  to  $T_i - 1$  **do**
- 6:        $x_t = \text{Encoder}(o_t)$
- 7:       Predict  $(\hat{a}_{t:t+k-1}, h_t) = \pi_\theta.\text{step}(x_t, h_{t-1})$
- 8:       Get  $a_{t:t+k-1}$  from  $\tau_i$
- 9:       Calculate step loss:  $\mathcal{L}_t = \mathcal{L}(\hat{a}_{t:t+k-1}, a_{t:t+k-1})$
- 10:       Accumulate loss:  $\mathcal{L}_{traj} += \mathcal{L}_t$
- 11:     **end for**
- 12:      $Opt.\text{step}()$             $\triangleright$  Update  $\theta$
- 13:   **end for**
- 14: **end for**
- 15: **return** Trained policy  $\pi_\theta$ .

---



---

#### Algorithm 2: MTIL Inference (with Action Chunking and Temporal Aggregation).

---

**Require:** Trained policy  $\pi_\theta$ , Initial observation  $o_0$ , Chunk size  $k$ , Max steps  $T_{max}$ , Exponential aggregation weights  $W$

- 1: Initialize hidden state  $h_0$ , prediction buffer  $B$ .
- 2: **for**  $t = 0$  to  $T_{max} - 1$  **do**
- 3:    $x_t = \text{Encoder}(o_t)$
- 4:   Predict  $(\hat{a}_{t:t+k-1}, h_t) = \pi_\theta.\text{step}(x_t, h_{t-1})$
- 5:   Store prediction  $\hat{a}_{t:t+k-1}$  in buffer  $B$
- 6:   Aggregate Action for step  $t$ :
- 7:     Get predictions for step  $t$  from  $B$ :  
 $P_t = \{\hat{a}_{j:j+k-1}[t-j] \mid j \leq t < j+k \text{ and } \hat{a}_{j:j+k-1} \in B\}$
- 8:     Compute final action:  
 $a_t^{\text{final}} = \text{WeightedAverage}(P_t, W)$
- 9:     Execute action  $a_t^{\text{final}}$
- 10: **end for**

---

selective mechanism, combined with inherent linear-time complexity  $O(L)$ , facilitates learning from *complete* trajectory histories—a significant advantage over quadratic-complexity  $O(L^2)$  attention mechanisms. The resulting state  $h_t$  acts as a dynamic summary of the salient history  $(x_1, \dots, x_{t-1})$ , furnishing the requisite context even when the instantaneous observation  $x_t$  is ambiguous. The MTIL policy leverages this directly:

$$\pi(\hat{a}_{t:t+K-1} | x_t, h_t)$$

By conditioning predictions  $\hat{a}_{t:t+K-1}$  on both the current input  $x_t$  and the comprehensive historical summary encoded in  $h_t$ , MTIL effectively transcends the limitations inherent in Markovian approaches, enabling sequential decision-making grounded in the full trajectory context.

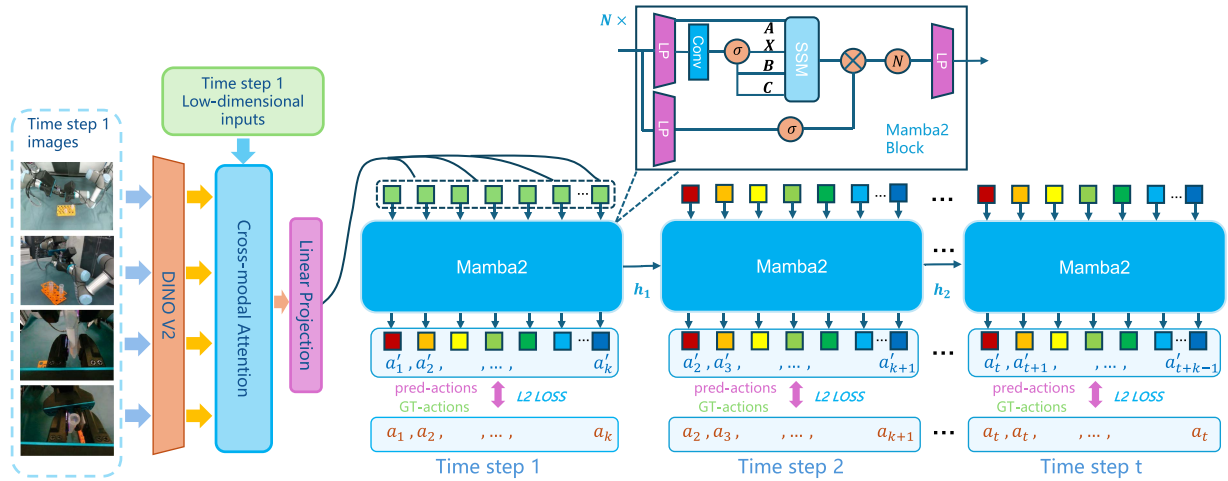


Fig. 1. Overview of the Mamba Temporal Imitation Learning (MTIL) architecture. Multi-modal inputs (images via DINOv2, state) are fused and processed by sequential Mamba-2 blocks, updating the recurrent state  $h_t$  which encodes history. At each step  $t$  across the entire trajectory, MTIL predicts an action chunk  $\hat{a}_{t:t+K-1}$  (current plus  $K-1$  future steps). This is supervised via L2 loss against ground truth actions  $a_{t:t+K-1}$  from the demonstration (using last action for padding when near trajectory end). The historical context embedded in  $h_t$  enables temporally coherent, long-horizon action generation.

### C. MTIL Training and Inference

MTIL enables imitation learning across complete expert trajectories, utilizing the architecture outlined in Fig. 1. Distinctively, MTIL employs a sequential training procedure (Algorithm 1). This step-wise paradigm, leveraging Mamba’s recurrent ‘step’ function, is fundamental to efficiently encoding arbitrarily long trajectories from high-dimensional observations (e.g., images) within feasible memory constraints—a key departure from parallel window-based approaches. A naive implementation of this sequential process would be limited to a batch size of one, posing a challenge for training efficiency. To address this, we introduce a novel **batch-parallel training scheme**. Instead of processing a single trajectory, our method processes a batch of  $B$  trajectories simultaneously. At each timestep  $t$ , the model takes a batch of observations, updates their respective hidden states and computes the loss concurrently. This approach preserves the crucial temporal integrity within each trajectory while fully leveraging the parallel processing power of modern GPUs, making MTIL’s training time competitive with highly-parallelizable Markovian methods. During training, at each timestep  $t$ , the policy receives the observation embedding  $x_t$ , updates its history-encoding state from  $h_{t-1}$  to  $h_t$ , and predicts an action chunk  $\hat{a}_{t:t+K-1}$ . Learning proceeds by minimizing the Mean Squared Error (MSE) against the ground truth actions  $a_{t:t+K-1}$ . During inference (Algorithm 2), the trained policy operates autoregressively, using the same ‘step’ function to update its state and predict actions. For enhanced stability, temporal aggregation strategies [5], [19], [20] are applied, averaging over predictions from overlapping action chunks to produce a smoother final action.

## IV. EXPERIMENTAL RESULTS

We conducted extensive experiments to evaluate the performance of MTIL across various benchmarks and real-world scenarios. All results stem from a rigorous protocol over three

TABLE I  
SUCCESS RATES (%) ON THE ACT BENCHMARK. RESULTS ARE AVERAGED OVER 3 SEEDS, ALL EXPERIMENTS RUN ON A SINGLE RTX 4090.

Method	History Length	Cube Transfer (%)	Bimanual Insertion (%)
ACT [5]	1 (Markovian)	90.0 ± 2.0	50.0 ± 3.5
Diffusion Policy [7]	1 (Markovian)	72.0 ± 2.6	28.0 ± 3.2
Diffusion Policy [7]	10	78.0 ± 2.5	32.0 ± 4.1
Diffusion Policy [7]	20	80.0 ± 2.2	34.0 ± 3.8
Diffusion Policy [7]	30	82.0 ± 2.1	36.0 ± 3.5
Diffusion Policy [7]	40		OOM
Transformer-XL [32]	Full (400)	86.0 ± 2.5	42.0 ± 4.0
MTIL (10-step)	10	92.0 ± 1.5	56.0 ± 2.5
<b>MTIL (Full)</b>	<b>Full (400)</b>	<b>100.0 ± 0.0</b>	<b>84.0 ± 2.1</b>

random seeds (100, 200, 300), with 50 roll-outs and the checkpoint for each run selected based on the minimum validation loss or as the final success rate for Robomimic.

### A. ACT Benchmark

We evaluated MTIL on the ACT benchmark to dissect its performance, efficiency, and learning dynamics against prominent architectural paradigms. The results, which juxtapose success rates with architectural choices and history lengths, are presented in Table I. The findings decisively establish MTIL’s superiority. On both tasks, MTIL (Full) achieves a perfect or near-perfect success rate, drastically outperforming all baselines. The learning curves in Fig. 2(a) illuminate this outcome, showing that MTIL not only attains a higher performance ceiling but also converges significantly faster, indicating a more stable and sample-efficient learning process. Conversely, the performance of attention-based models reveals a critical insight: naively incorporating history is an inefficient, and ultimately, a computationally infeasible strategy. While Diffusion Policy’s success rate scales with history length, it remains notably inferior to the simple Markovian ACT baseline and incurs a prohibitive computational cost, culminating in an Out-of-Memory (OOM) error. Even Transformer-XL, theoretically capable of

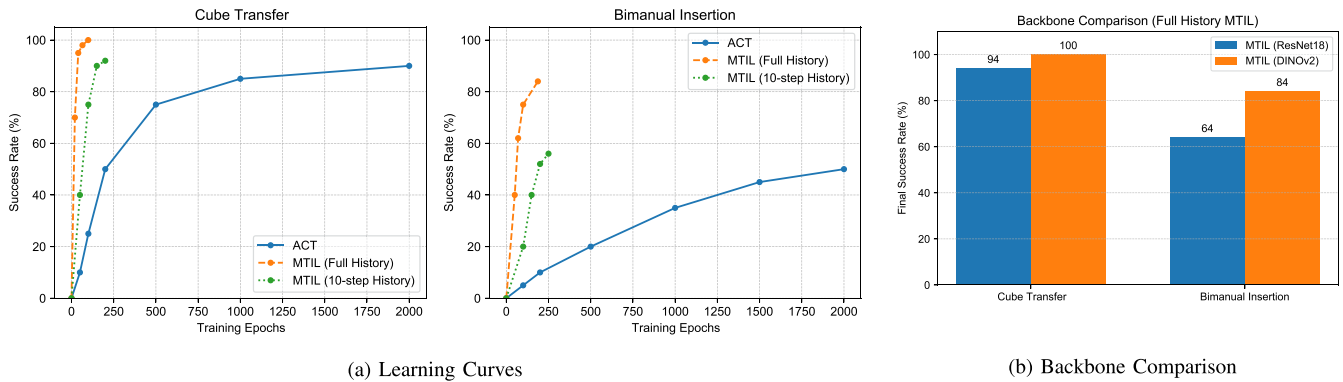


Fig. 2. ACT benchmark performance. (a) Learning curves (Cube Transfer left, Bimanual Insertion right). (b) Backbone comparison (MTIL Full-History, DINOv2 vs. ResNet18).

TABLE II  
LIFELONG LEARNING PERFORMANCE ON LIBERO (EWC STRATEGY)

Policy Arch.	LIBERO-LONG			LIBERO-SPATIAL		
	FWT( $\uparrow$ )	NBT( $\downarrow$ )	AUC( $\uparrow$ )	FWT( $\uparrow$ )	NBT( $\downarrow$ )	AUC( $\uparrow$ )
RESNET-RNN	0.02 $\pm$ 0.00	0.04 $\pm$ 0.01	0.00 $\pm$ 0.00	0.14 $\pm$ 0.02	0.23 $\pm$ 0.02	0.03 $\pm$ 0.00
RESNET-T	0.13 $\pm$ 0.02	0.22 $\pm$ 0.03	0.03 $\pm$ 0.00	0.23 $\pm$ 0.01	0.33 $\pm$ 0.01	0.06 $\pm$ 0.01
RESNET-M (10-STEP)	0.14 $\pm$ 0.02	0.20 $\pm$ 0.03	0.03 $\pm$ 0.00	0.24 $\pm$ 0.01	0.30 $\pm$ 0.02	0.06 $\pm$ 0.01
RESNET-M (FULL)	<b>0.22 <math>\pm</math> 0.03</b>	0.08 $\pm$ 0.02	0.08 $\pm$ 0.02	0.28 $\pm$ 0.02	0.17 $\pm$ 0.02	0.05 $\pm$ 0.01
ViT-T	0.05 $\pm$ 0.02	0.09 $\pm$ 0.03	0.01 $\pm$ 0.00	0.32 $\pm$ 0.01	0.48 $\pm$ 0.03	0.06 $\pm$ 0.01
ViT-M (10-STEP)	0.06 $\pm$ 0.02	0.10 $\pm$ 0.03	0.01 $\pm$ 0.00	0.33 $\pm$ 0.01	0.45 $\pm$ 0.03	0.06 $\pm$ 0.01
ViT-M (FULL)	0.19 $\pm$ 0.04	<b>0.05 <math>\pm</math> 0.01</b>	<b>0.10 <math>\pm</math> 0.03</b>	<b>0.35 <math>\pm</math> 0.02</b>	<b>0.15 <math>\pm</math> 0.03</b>	<b>0.10 <math>\pm</math> 0.01</b>
Policy Arch.	LIBERO-OBJECT			LIBERO-GOAL		
	FWT( $\uparrow$ )	NBT( $\downarrow$ )	AUC( $\uparrow$ )	FWT( $\uparrow$ )	NBT( $\downarrow$ )	AUC( $\uparrow$ )
RESNET-RNN	0.17 $\pm$ 0.04	0.23 $\pm$ 0.04	0.06 $\pm$ 0.01	0.16 $\pm$ 0.01	0.22 $\pm$ 0.01	0.06 $\pm$ 0.01
RESNET-T	0.56 $\pm$ 0.03	0.69 $\pm$ 0.02	0.16 $\pm$ 0.02	0.32 $\pm$ 0.04	0.45 $\pm$ 0.04	0.07 $\pm$ 0.01
RESNET-M (10-STEP)	0.50 $\pm$ 0.03	0.39 $\pm$ 0.03	0.15 $\pm$ 0.02	0.31 $\pm$ 0.04	0.42 $\pm$ 0.04	0.07 $\pm$ 0.01
RESNET-M (FULL)	0.55 $\pm$ 0.03	0.36 $\pm$ 0.03	0.17 $\pm$ 0.01	0.30 $\pm$ 0.03	0.11 $\pm$ 0.04	0.10 $\pm$ 0.01
ViT-T	0.57 $\pm$ 0.03	0.64 $\pm$ 0.03	0.23 $\pm$ 0.00	0.32 $\pm$ 0.04	0.48 $\pm$ 0.03	0.07 $\pm$ 0.01
ViT-M (10-STEP)	0.56 $\pm$ 0.03	0.60 $\pm$ 0.03	0.22 $\pm$ 0.01	0.33 $\pm$ 0.04	0.45 $\pm$ 0.03	0.08 $\pm$ 0.01
ViT-M (FULL)	<b>0.58 <math>\pm</math> 0.03</b>	<b>0.18 <math>\pm</math> 0.04</b>	<b>0.25 <math>\pm</math> 0.01</b>	<b>0.34 <math>\pm</math> 0.04</b>	<b>0.10 <math>\pm</math> 0.03</b>	<b>0.11 <math>\pm</math> 0.01</b>

full-history processing, fails to match ACT, reinforcing the hypothesis that attention is a suboptimal inductive bias for modeling the continuous dynamics of physical interaction. Furthermore, the backbone ablation in Fig. 2(b) confirms our advantage is architectural. MTIL, even with an identical ResNet18 backbone [50], substantially outperforms ACT. The use of a stronger DINOv2 backbone [51] further widens this gap. This proves MTIL’s success stems from a fundamentally superior paradigm: a computationally efficient recurrent architecture that is intrinsically better suited to capturing the temporal fabric of the physical world.

## B. LIBERO Benchmark

On LIBERO’s [52] EWC [53] lifelong learning benchmark (using standard ResNet/ViT backbones matching baselines for fair comparison), MTIL demonstrates strong lifelong learning when leveraging full history (-M (FULL), Table II). It consistently achieves superior forward transfer (FWT $\uparrow$ ), reduced forgetting (NBT $\downarrow$ ), and higher overall performance (AUC $\uparrow$ ) compared to baselines and short-history (10-step) MTIL, which performs similarly to Transformers (-T).

This advantage of full-history encoding, while notable across all categories, becomes particularly pronounced in **LIBERO-LONG**. Here, the performance margin over limited-context methods widens substantially, offering compelling evidence for the critical role of complete history as task horizons extend.

## C. Robomimic (Vision-Based Policy)

To assess MTIL’s ability to handle high-dimensional visual inputs, we evaluated it on the vision-based Robomimic tasks [12]. As shown in Table III, MTIL (Full) significantly outperforms all baselines, including the strong DiffusionPolicy variants. Notably, MTIL (10-step) offers only a marginal improvement over the LSTM-GMM baseline, highlighting that a short-history SSM is insufficient. The substantial performance gain of MTIL (Full) underscores its superior capability in leveraging full spatio-temporal context from visual data. This can be attributed to its nature as a learned dynamical system; the recurrent state  $h_t$  acts as an implicit world model, tracking not just object locations but also their latent physical states (e.g., momentum, contact stability) over time, which is crucial for complex manipulation.

TABLE III

BEHAVIOR CLONING BENCHMARK (VISUAL POLICY) ON ROBOMIMIC. AS PER THE ORIGINAL DATASET, RESULTS ARE REPORTED AS FINAL SUCCESS RATES.

	Lift		Can		Square		Transport		ToolHang
	ph	mh	ph	mh	ph	mh	ph	mh	ph
LSTM-GMM [29]	1.00	1.00	1.00	0.98	0.82	0.64	0.88	0.44	0.68
IBC [12]	0.94	0.39	0.08	0.00	0.03	0.00	0.00	0.00	0.00
DiffusionPolicy-C	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.98	<b>0.98</b>	<b>1.00</b>	0.89	0.95
DiffusionPolicy-T	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	1.00	0.94	0.98	0.73	0.76
MTIL (10-step)	1.00	1.00	1.00	0.99	0.87	0.65	0.92	0.52	0.72
<b>MTIL (Full)</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.96	<b>1.00</b>	<b>0.91</b>	<b>0.97</b>



Fig. 3. Dual UR3 experimental setup with four cameras (Top: Kinect; Side: D435i; Wrists: D405) and custom grippers.

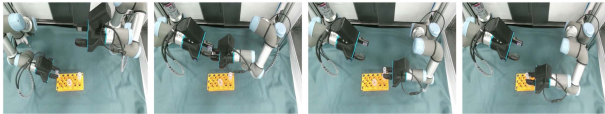


Fig. 4. Sequential Insertion task stages (left panel of figure) and success rates (right panel, Table IV). MTIL successfully completes the sequence while ACT fails due to temporal ambiguity between stages 3 and 4.

TABLE IV

SEQUENTIAL INSERTION SUCCESS RATES (%), AVERAGED OVER 50 ROLL-OUTS

Method	Stage 1	Stage 2	Stage 3	Stage 4	Overall
ACT	80.0±4.5	64.0±5.1	0.0±0.0	0.0±0.0	0.0±0.0
<b>MTIL</b>	<b>94.0±2.5</b>	<b>80.0±4.0</b>	<b>62.0±5.8</b>	<b>54.0±4.5</b>	<b>54.0±4.5</b>

#### D. Real-World Dual-Arm Tasks

To validate MTIL in complex physical environments, we designed challenging tasks on a dual UR3 platform equipped with custom 2-finger grippers and four cameras providing multi-view observations (Fig. 3). We compare MTIL (using DINOv2 backbone and full history) against ACT trained on identical demonstration data (100 demos per task). All real-world results are averaged over 50 evaluation roll-outs for the best checkpoint from each of the 3 seeds.

*a) Sequential Insertion Task:* We designed this task (visualized in Fig. 4) specifically to challenge Markovian policies by requiring long-term memory, a scenario where SOTA methods like ACT often fail. The four stages involve: (1) Left arm grasps Tube1, (2) Left passes Tube1 to Right arm, (3) Right arm inserts Tube1 into Tube2, (4) Right arm inserts Tube1 into Tube3. Critically, executing Stage 3 correctly necessitates recalling the completion of previous stages, as intermediate observations can be ambiguous. Table IV details the stage-wise success rates. MTIL, leveraging its full history state, successfully completes



Fig. 5. Coordinated Pouring task stages (left panel of figure) and success rates (right panel, Table V). MTIL achieves higher success and smoother execution compared to ACT.

TABLE V

COORDINATED POURING SUCCESS RATES (%), AVERAGED OVER 50 ROLL-OUTS

Method	Stage 1	Stage 2	Stage 3	Stage 4	Overall
ACT	80.0±4.5	64.0±5.1	50.0±6.3	32.0±5.0	32.0±5.0
<b>MTIL</b>	<b>94.0±2.5</b>	<b>80.0±4.0</b>	<b>74.0±4.8</b>	<b>62.0±5.8</b>	<b>62.0±5.8</b>

the entire sequence with high probability. In stark contrast, ACT, reliant on immediate context, is confounded by the temporal ambiguity, as the observations after completing Stage 2 can be identical with completing Stage 3, making it indistinguishable for policies relying solely on current or short-term history. AS a result, it frequently attempts Stage 4 directly after Stage 2, failing to execute the required sequence correctly and resulting in zero success for completing Stage 3, Stage 4, and the overall task. This outcome underscores the limitations of short-history approaches and validates the imperative of encoding complete history for reliably executing temporally complex manipulation sequences.

*b) Coordinated Pouring Task:* This task (Fig. 5) assesses precise bimanual coordination over a longer sequence: (1) Left arm grasps Tube1, (2) Left passes Tube1 to Right arm, (3) Left arm grasps Tube2, (4) Right arm pours water from Tube1 into Tube2. While less susceptible to the specific ambiguity of the insertion task, it still requires accurate, temporally coordinated actions. Table V (within Fig. 5) shows that although both methods achieve non-zero success, MTIL consistently outperforms ACT across the stages, resulting in a higher overall success rate and exhibiting notably smoother execution trajectories.

#### V. CONCLUSION

The trajectory of intelligence is intrinsically linked to the capacity for memory – the ability to weave the tapestry of past experiences into the fabric of present action. This work confronts a central limitation in contemporary imitation learning: the prevalent reliance on the Markovian assumption, which often reduces complex sequential behaviors to mere reactions to the immediate sensory world. We introduced Mamba Temporal Imitation Learning (MTIL), a new paradigm that embraces the power of memory by leveraging the recurrent state dynamics inherent within the Mamba architecture. We posit that MTIL represents a practical and powerful synthesis of concepts from World Models and Dynamical Systems. By encoding the full history of observations into a compressed, evolving state representation, MTIL learns an implicit, action-oriented world model. This comprehensive temporal context allows MTIL to effectively disambiguate perception and unlock the execution of intricate, state-dependent sequential tasks previously challenging for established methods. Our findings not only showcase the

significant performance and efficiency gains afforded by MTIL but, more profoundly, underscore the essential role of history in bridging the gap between perception and intelligent action. By demonstrating the efficacy of SSMs in capturing the long flow of time in a computationally feasible manner, this work illuminates a promising pathway towards building robotic agents capable of deeper understanding and more sophisticated interaction with the world.

## REFERENCES

- [1] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Found. Trends Robot.*, vol. 7, no. 1/2, pp. 1–179, 2018.
- [2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.
- [3] S. Schaal, "Learning from demonstration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 1040–1046.
- [4] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 1989, pp. 305–313.
- [5] T. Z. Zhao et al., "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. Robot., Sci. Syst.*, Daegu, South Korea, 2023.
- [6] T. Z. Zhao et al., "Aloha unleashed: A simple recipe for robot dexterity," in *Proc. 8th Annu. Conf. Robot Learn.*, 2024, pp. 1–18.
- [7] C. Chi et al., "Diffusion Policy: Visuomotor policy learning via action diffusion," in *Proc. Robot., Sci. Syst.*, Daegu, South Korea, 2023.
- [8] J. Ho, A. Jain, and P. Abbeel, "Denosing diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6840–6851.
- [9] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D diffusion policy," 2024, *arXiv:2403.03954*.
- [10] T. Pearce et al., "Imitating human behaviour with diffusion models," in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
- [11] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statistics. JMLR Workshop Conf. Proc.*, 2011, pp. 627–635.
- [12] A. Mandlekar et al., "What matters in learning from offline human demonstrations for robot manipulation," in *Proc. Conf. Robot Learn.*, 2021, pp. 950–961.
- [13] L. Chen et al., "Decision Transformer: Reinforcement learning via sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 15084–15097.
- [14] C. Lynch et al., "Learning latent plans from play," in *Proc. Conf. Robot Learn.*, 2020, pp. 1088–1103.
- [15] T. Gao, S. Nasiriany, H. Liu, Q. Yang, and Y. Zhu, "PRIME: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning," *IEEE Robot. Automat. Lett.*, vol. 9, no. 10, pp. 8322–8329, Oct. 2024.
- [16] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2023, *arXiv:2312.00752*.
- [17] X. Jia et al., "MaIL: Improving imitation learning with selective state space models," in *Proc. 8th Annu. Conf. Robot Learn.*, 2024, pp. 1–17.
- [18] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [19] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, "Waypoint-based imitation learning for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2023, pp. 2195–2209.
- [20] X. Zhang, Y. Liu, H. Chang, L. Schramm, and A. Boularias, "Autoregressive action sequence learning for robotic manipulation," *IEEE Robot. Automat. Lett.*, vol. 10, no. 5, pp. 4898–4905, May 2025.
- [21] X. Liu, Y. Zhou, F. Weigend, S. Sonawani, S. Ikemoto, and H. B. Amor, "Diff-control: A stateful diffusion-based policy for imitation learning," in *Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7453–7460.
- [22] P. Florence et al., "Implicit behavioral cloning," in *Proc. Conf. Robot Learn.*, 2022, pp. 154–167.
- [23] M. A. Bashiri, B. Ziebart, and X. Zhang, "Distributionally robust imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 24404–24417.
- [24] M. Beck et al., "xLSTM: Extended long short-term memory," in *Proc. Adv. Neural Inf. Process. Syst.*, 2025, vol. 37, pp. 107547–107603.
- [25] A. Mandlekar et al., "MimicGen: A data generation system for scalable robot learning using human demonstrations," in *Proc. 7th Annu. Conf. Robot Learn.*, 2023, pp. 1–16.
- [26] N. M. Shafullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior Transformers: Cloning  $k$  modes with one stone," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 22955–22968.
- [27] A. Brohan et al., "Rt-1: Robotics transformer for real-world control at scale," 2022, *arXiv:2212.06817*.
- [28] S. Haldar, Z. Peng, and L. Pinto, "BAKU: An efficient transformer for multi-task policy learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.
- [29] M. Dalal et al., "Imitating task and motion planning with visuomotor transformers," in *Proc. 7th Annu. Conf. Robot Learn.*, 2023, pp. 1–15.
- [30] L. Fu et al., "In-context imitation learning via next-token prediction," in *Proc. NeurIPS Workshop Open-World Agents*, 2024.
- [31] M. Reuss, Ö. E. Yagmurlu, F. Wenzel, and R. Lioutikov, "Multimodal diffusion transformer: Learning versatile behavior from multimodal goals," 2024, *arXiv:2407.05996*.
- [32] G. Tianci, "Transformer-xl for long sequence tasks in robotic learning from demonstrations," 2024, *arXiv:2405.15562*.
- [33] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4126–4133, Apr. 2022.
- [34] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," in *Proc. Conf. Robot Learn.*, 2019, pp. 1001–1013.
- [35] W. Mao et al., "Robomatrix: A skill-centric hierarchical framework for scalable robot task planning and execution in open-world," 2024, *arXiv:2412.00171*.
- [36] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, "Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization," in *Proc. 5th Annu. Conf. Robot Learn.*, 2021, pp. 1–11.
- [37] Z. Chen, Z. Ji, J. Huo, and Y. Gao, "Scar: Refining skill chaining for long-horizon robotic manipulation via dual regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 111679–111714.
- [38] P. Bevanda, M. Beier, A. Capone, S. G. Sosnowski, S. Hirche, and A. Lederer, "Koopman-equivariant gaussian processes," in *Proc. 28th Int. Conf. Artif. Intell. Statist.*, 2025.
- [39] T. Fernando and M. Darouach, "Existence and design of target output controllers," *IEEE Trans. Autom. Control*, vol. 70, no. 9, pp. 6104–6110, Sep. 2025.
- [40] J. Luo, J. Cheng, X. Tang, Q. Zhang, B. Xue, and R. Fan, "Mambaflow: A novel and flow-guided state space model for scene flow estimation," 2025, *arXiv:2502.16907*.
- [41] J. Du, Y. Sun, Z. Zhou, P. Chen, R. Zhang, and K. Mao, "Mambaflow: A mamba-centric architecture for end-to-end optical flow estimation," 2025, *arXiv:2503.07046*.
- [42] K. Zeng et al., "Mambamos: Lidar-based 3D moving object segmentation with motion-aware state space model," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 1505–1513.
- [43] T. Tsuji, "Mamba as a motion encoder for robotic imitation learning," *IEEE Access*, vol. 13, pp. 69941–69949, 2025.
- [44] X. Jia et al., "X-il: Exploring the design space of imitation learning policies," 2025, *arXiv:2502.12330*.
- [45] J. Cao et al., "Mamba policy: Towards efficient 3D diffusion policy with hybrid selective state models," 2024, *arXiv:2409.07163*.
- [46] M. Reuss, J. Pari, P. Agrawal, and R. Lioutikov, "Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning," 2024, *arXiv:2412.12953*.
- [47] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1/2, pp. 99–134, 1998.
- [48] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *Proc. 10th Int. Conf. Learn. Representations*, 2022.
- [49] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 10041–10071.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [51] M. Oquab et al., "Dinov2: Learning robust visual features without supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023.
- [52] B. Liu et al., "Libero: Benchmarking knowledge transfer for lifelong robot learning," in *Proc. Adv. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2023, pp. 44776–44791.
- [53] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, pp. 3521–3526, 2017.