Lifelong Imitation Learning with Multimodal Latent Replay and Incremental Adjustment

Fanqi Yu[1,3] Matteo Tiezzi[4†] Tommaso Apicella[4†] Cigdem Beyan[1,2] Vittorio Murino[1,2]
[1]AI for Good (AIGO), Istituto Italiano di Tecnologia, Genoa, Italy
[2]Department of Computer Science, University of Verona, Verona, Italy
[3]DITEN, University of Genoa, Genoa, Italy
[4]PAVIS, Istituto Italiano di Tecnologia, Genoa, Italy
[†]These authors contributed equally as second authors.

Abstract

We introduce a lifelong imitation learning framework that enables continual policy refinement across sequential tasks under realistic memory and data constraints. Our approach departs from conventional experience replay by operating entirely in a multimodal latent space, where compact representations of visual, linguistic, and robot's state information are stored and reused to support future learning. To further stabilize adaptation, we introduce an incremental feature adjustment mechanism that regularizes the evolution of task embeddings through an angular margin constraint, preserving inter-task distinctiveness. Our method establishes a new state of the art in the LIBERO benchmarks, achieving 10–17 point gains in AUC and up to 65% less forgetting compared to previous leading methods. Ablation studies confirm the effectiveness of each component, showing consistent gains over alternative strategies. The code is available at: https://github.com/yfqi/lifelong_mlr_ifa.

1. Introduction

Imitation Learning (IL) enables agents, such as robots, to learn behaviors by observing and mimicking human demonstrations [7, 8, 12, 24, 27]. However, real-world environments are dynamic, with new objects, goals, and contexts constantly emerging. For example, a household robot may encounter unfamiliar kitchen tools, rearranged furniture, or previously unseen tasks. To operate effectively, agents must continuously learn and adapt throughout their lifetime. Standard IL typically assumes a fixed set of tasks and does not account for new tasks or variations in the environment [30, 31]. Lifelong Imitation Learning (LIL) addresses this limitation by enabling agents to continuously acquire new skills while retaining previously learned behaviors, even as the set of tasks evolves [14, 25, 29, 32]. LIL focuses on preventing catastrophic forgetting [9], allowing knowledge from earlier tasks to be reused for new ones, and supports learning from an unbounded or dynamically growing sequence of tasks, rather than a fixed predefined set [2, 4, 23, 26].

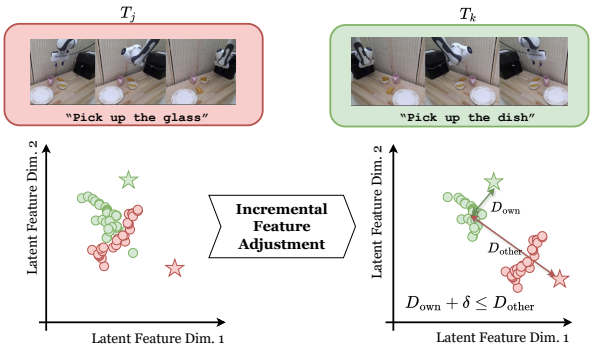


Figure 1. Illustration of Incremental Feature Adjustment (IFA). The figure displays a 2D projection of the global latent representations $\mathbf{g}$ during policy rollout for two related tasks, T_j (previously learned) and T_k (newly learned). The stars ($\star$ for T_j and $\star$ for T_k) represent the stable language reference embeddings $\mathbf{h}^{(r)}$, while the circles ($\bullet$ for T_j and $\bullet$ for T_k) are the global embeddings $\mathbf{g}$. (Left) Without IFA, the new task's embeddings ($\bullet$) exhibit representation drift by clustering close to the old task's embeddings ($\bullet$). (Right) With IFA, the loss $\mathcal{L}_{\text{IFA}}$ enforces a constraint on the distances: the distance to the own reference (D_{own}) plus a margin δ must be less than or equal to the distance to the other task's reference (D_{other}). This mechanism forces the $\mathbf{g}(T_k)$ ($\bullet$) cluster away from $\mathbf{h}^{(r)}(T_j)$ ($\star$) and closer to $\mathbf{h}^{(r)}(T_k)$ ($\star$) , achieving inter-task disentanglement.

However, many practical methods rely on additional assumptions to prevent forgetting. For instance, some approaches assume that the task identifier is available during both training and evaluation, allowing task-specific networks or adapters to isolate knowledge for each task [15].

In this way, these methods retain performance on previously learned tasks while sequentially acquiring new ones. In contrast, fully task-agnostic LIL methods do not require task identifiers at test-time (referred to as *task-ID agnostic* throughout this paper) and typically rely on strategies such as experience replay (also called rehearsal-based) [4, 14, 26], or multi-task distillation [23].

Recent LIL methods have increasingly leveraged large pretrained models, such as Vision-Language Models (e.g., CLIP [21]) and Large Language Models (e.g., GPT [1]). However, directly using these models in their pretrained form on natural images or text is often insufficient for complex imitation learning tasks. To address this, several approaches, e.g., [10, 15, 26] adopt a two-stage paradigm: a pretraining stage (also called the base task stage) followed by a lifelong learning stage. This strategy leverages the rich representations learned during pretraining while enabling continuous adaptation to new tasks. During the lifelong learning stage, some prior methods still employ parameter-efficient fine-tuning (PEFT) (e.g., [3, 13, 28]) to adapt encoders to novel tasks [10, 15].

In this study, we introduce a rehearsal-based method that applies a Multimodal Latent Replay (MLR), which stores joint compact latent representations that encapsulate vision, language, state (e.g., robot's orientation, position) modalities with control commands. When a new task is encountered, its latent representations may overlap those of previously learned tasks [6], leading to interference in the shared embedding space. To address this, we propose Incremental Feature Adjustment (IFA), a representation-level regularization strategy that maintains reference embeddings for past tasks, such as language-based task embeddings or centroids of aggregated feature representations (See Fig. 1). IFA is a loss on angular distances to encourage the current task's representation to remain distinct from previous tasks while maintaining alignment with its own reference. The loss margin is adaptively scaled based on inter-task similarity, reducing sensitivity to dataset-specific feature variations and eliminating the need for manual tuning. By progressively repelling the current task from old-task anchors, IFA preserves inter-task separability and mitigates interference during lifelong learning.

Our approach uses a frozen CLIP encoder [21], demonstrating that effective LIL can be achieved without fine-tuning the backbone during the lifelong learning stage, unlike [10, 15], which rely on PEFT (e.g., vision adapters). Different from rehearsal-based LIL methods, e.g., [14, 26], which store raw data (e.g., scenes and trajectories), we employ compact MLR, which is memory-efficient. By combining MLR with IFA, our method achieves robust lifelong learning in a task-ID agnostic manner, offering a simpler alternative to distillation [23] or generative approaches [4].

Experimental analysis on three different datasets demonstrates that the proposed MLR and IFA modules provide noticeable improvements over existing state-of-the-art (SOTA). Ablation studies show that each component contributes meaningfully, outperforming alternative designs or variations built on different assumptions.

The contributions of this work can be summarized as:

- A multimodal latent replay framework for learning new tasks using a model with pre-trained unimodal encoders (visual, language, and state). By storing latent features instead of raw data, our method reduces the memory footprint of previously learned tasks and mitigates forgetting.
- An Incremental Feature Adjustment module that separates the latent representations of old and new tasks. Specifically, we modulate the margin between representations based on the angular distance between old and new task embeddings, adapting the strength of the loss according to the semantic similarity between tasks.

2. Related Work

Lifelong Imitation Learning (LIL) aims to enable robotic agents to acquire new skills over time without forgetting previously learned ones, a challenge commonly referred to as catastrophic forgetting. To address this problem, some studies have employed rehearsal-based techniques (also known as experience replay) [4, 14, 26], which replay stored past data to help the model maintain performance on earlier tasks. For instance, [14], building upon ER [2], maintains a selection of past trajectories and interleaves them with new ones from the current task during training. In comparison, CRIL [4] employs generative adversarial networks (GANs) to synthesize the first frame of each trajectory and uses an action-conditioned video prediction model to generate subsequent frames based on the current states and actions. LOTUS [5] allows robots to incrementally acquire new skills by identifying recurring patterns within unsegmented demonstrations. These skills are extracted via an open-vocabulary vision model and orchestrated by a meta-controller, enabling the agent to solve complex, long-horizon manipulation tasks. Despite their effectiveness, rehearsal-based approaches are often sensitive to the replay ratio and the similarity between new and previously learned tasks, resulting in variable levels of forgetting.

Another line of work focuses on progressive model expansion, where additional parameters or modules are introduced as new tasks arrive, enabling the architecture to grow adaptively and incorporate new knowledge. BUDS [33] proposes a method for discovering reusable skills in robotic manipulation without relying on pre-segmented demonstrations. Using a bottom-up strategy, it autonomously identifies and organizes skills from long-horizon, unsegmented demonstrations, allowing robots to handle complex manipulation tasks effectively. In contrast, TAIL [15] applies several Parameter-Efficient Fine-Tuning (PEFT) methods, such

as LoRA, to CLIP encoders, creating a specific adapter for each new task. A key limitation of this approach is that it requires the task identifier at test time, which may not always be available in practical scenarios. Another adapter-based method, ISCIL [10], addresses this limitation by promoting knowledge sharing through the incremental learning of reusable skills across multiple demonstrations. These skills are stored in a prototype-based memory, enabling sample-efficient task adaptation and better generalization in non-stationary LIL environments. Overall, progressive approaches generally rely on explicit stage or task identification during evaluation and often face challenges in generalizing to unseen tasks.

Differently, M2Distill [23] is a distillation-based LIL method that preserves a consistent latent space across vision, language, and action modalities as new tasks are learned. It regulates distribution shifts between consecutive learning steps via multimodal distillation and Gaussian Mixture Model (GMM) policy alignment, ensuring that previously acquired skills are retained while new ones are integrated efficiently.

Our approach can be categorized as rehearsal-based, but it differs from the above-mentioned methods by replaying multimodal latent representations, which are the compositions of vision, language, the state of the robot, and action modalities. Considering that high similarity between new and previously learned tasks can degrade the performance of standard rehearsal-based methods, we introduce a incremental feature adjustment mechanism based on angular distance. Compared to other approaches, our pipeline is much simpler: it does not rely on knowledge distillation, does not use any PEFT methods, and instead operates on completely frozen backbones. Moreover, unlike methods such as LOTUS [26], our approach does not require an open-vocabulary vision encoder to achieve superior performance.

3. Methodology

3.1. Problem Formulation

We frame the LIL problem as consisting of two sequential phases: a *multi-task pre-training stage* and a *lifelong learning stage*, following prior work [10, 15, 26]. In this subsection, we provide an overview of this problem formulation, while the next subsection details our approach.

Multi-Task Pre-Training. Let $\mathcal{T}' = \{T'_j\}_{j=1}^{J}$ denote a set of pre-training tasks. Each task T'_j is associated with a set of N_j expert demonstrations (e.g., trajectories collected from human operators or expert policies that successfully perform the task) $\mathcal{D}_j = \{\tau_{j,i}\}_{i=1}^{N_j}$, where each trajectory $\tau_{j,i} = \{(o_t, a_t)\}_{t=1}^{L_j}$ consists of a sequence of L_j timesteps of observation-action pairs collected from expert rollouts. Here, the observation o_t typically includes sensory observations such as different views of images, proprioceptive signals and language description of the task to be performed, while a_t represents the expert's control command (ground truth action).

The policy $\pi_\theta(\hat{a}_t \mid o_{\leq t})$, parameterized by θ, predicts the action $\hat{a}_t$ conditioned on the most recent L_j observations, denoted with $o_{\leq t} \triangleq (o_{t-L_j+1}, \cdots, o_t)$, and is trained to imitate expert actions using a behavioral cloning (BC) objective [19]:

$$\min_{\theta} \sum_{j=1}^{J} \mathbb{E}_{(o_{\leq t}, a_t)\sim\mathcal{D}_j} \left[\mathcal{L}_{\text{BC}}\big(\pi_\theta(\hat{a}_t \mid o_{\leq t}),\, a_t\big) \right], \qquad (1)$$

where $\mathcal{L}_{\text{BC}}$ denotes a supervised imitation loss (e.g., negative log-likelihood or mean-squared error). This pre-training phase is performed jointly over all tasks in an *offline* multi-task setting, serving to establish shared representations and behavioral priors before lifelong learning begins.

Lifelong Learning. After the pre-training stage, the agent encounters a sequence of unseen tasks $\mathcal{T} = \{T_1, T_2, \ldots, T_K\}$, where each task T_k arrives sequentially, with $\mathcal{T} \cap \mathcal{T}' = \emptyset$. At task k, the agent has access to a set of expert demonstrations $\mathcal{D}_k = \{\tau_{k,i}\}_{i=1}^{N_k}$ from the current task, and a limited replay buffer $\mathcal{B}$. The goal is to adapt the policy π_θ on the union of current and replayed samples while avoiding catastrophic forgetting.

Formally, the learning objective at task k is defined as:

$$\min_{\theta} \mathbb{E}_{(o_{\leq t}, a_t)\sim(\mathcal{D}_k \cup \mathcal{B})} \left[\mathcal{L}_{\text{BC}}\big(\pi_\theta(\hat{a}_t \mid o_{\leq t}),\, a_t\big) \right]. \qquad (2)$$

3.2. Our Approach

We propose a LIL framework that introduces two complementary components for the lifelong learning stage: (1) Multimodal Latent Replay (MLR), which replays compact multimodal latent representations instead of raw trajectories, and (2) Incremental Feature Adjustment (IFA), which regularizes cross-task representations to maintain stability during continual adaptation. Below, we will first briefly describe the overall information flow in the neural architecture, including our pretraining, and then describe the main components we propose for the lifelong learning stage.

Base Policy Architecture. The policy π_θ processes multimodal observations $o_t = \{o_t^{(v)}, o^{(l)}, o_t^{(s)}\}$, corresponding to the modalities of visual, language , and state. Each modality is encoded into a latent feature $\mathbf{h}_t^{(m)} \in \mathbb{R}^E$, with $m \in \{v, l, s\}$ and where E denotes the embedding size, via modality-specific encoders, modulated by a proper network and concatenated over the past L time steps to form a multimodal sequence $\mathbf{H} \in \mathbb{R}^{M\times L\times E}$, where M denotes number of modalities and L the temporal length of the rollout, respectively. A temporal decoder maps $\mathbf{H}$ into a *global latent representation* $g_t \in \mathbb{R}^E$, which is then decoded by the policy head to predict the next action $\hat{a}_t$. During pre-training,

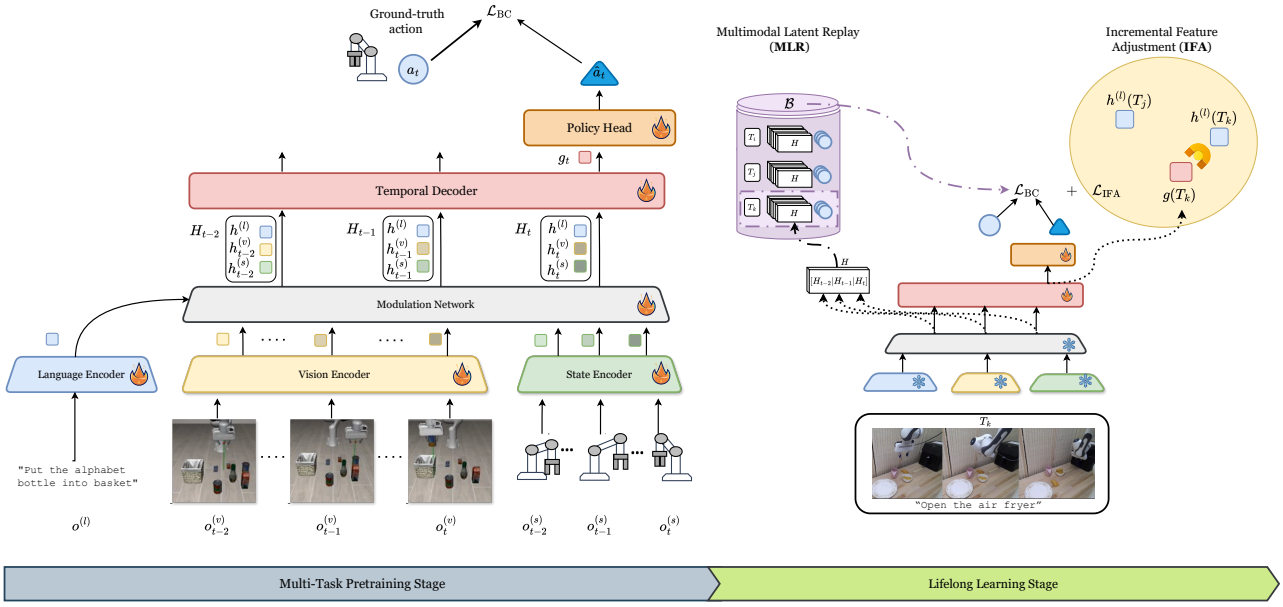


Figure 2. Our method is a general multimodal architecture composed of modality-specific encoders (language, vision, and state), a modulation network, a temporal decoder, and a policy head. During the pretraining phase, all architecture modules are trained. In the lifelong learning phase, only the temporal decoder and policy head are updated using both the new task data and the samples stochastically stored in the replay buffer. The buffer stores the multimodal features, output of the modulation layer. The model is jointly supervised by both the Behavior Cloning and the Incremental Feature Adjustment loss, processing the current task and previously stored tasks.

all modules are trainable. Fig. 2 illustrates the proposed architecture together with the training pipeline. The corresponding implementation details are described in the following text and in Sec. 3.3.

Multimodal Latent Replay (MLR). During the lifelong learning phase, only the temporal decoder and policy head are trainable; all other networks remain frozen as learned during pre-training. We leverage a replay-based method, MLR, that instead of storing past trajectories with the associated ground truth action $(o_{\leq t}, a_t)$ as in experience replay [14], maintains a compact buffer $\mathcal{B}$ of multimodal latent representations:

$$\mathcal{B} = \{(\mathbf{H}_n, a_n)\}_{n=1}^{N_B}, \tag{3}$$

where $\mathbf{H}_n$ is the concatenated latent produced by the frozen encoders, and N_B denotes the total buffer capacity. When learning a new task, replayed latents from $\mathcal{B}$ are combined with the current data $\mathcal{D}_k$ to form the imitation objective. MLR effectively stabilizes training and mitigates catastrophic forgetting without accessing raw sensory inputs (e.g., high-dimensional images that occupy large storage space, whereas latent trajectories are much smaller).

Incremental Feature Adjustment. To counter representation drift across tasks, we introduce *Incremental Feature Adjustment (IFA)*, which regularizes the relationship between latent representations of old and new tasks. Let $g_t(T_i)$ denote the global latent representation obtained from a demonstration of task T_i, and let $h^{(r)}(T_i)$ denote a reference embedding for the same task. When learning a new task T_k, we penalize configurations where $g_t(T_k)$ is closer to old task references than to its own reference:

$$\mathcal{L}_{\text{IFA}} = \frac{1}{|\mathcal{P}|} \sum_{\substack{(j,k)\in\mathcal{P} \\ j<k}} \max\Big(0, d\big(g_t(T_k), h^{(r)}(T_k)\big) - d\big(g_t(T_k), h^{(r)}(T_j)\big) + \delta\Big), \tag{4}$$

where $d(\cdot,\cdot)$ is a distance measure between two representations, $\mathcal{P}$ denotes the set of task pairs that are selected for adjustment, the loss is averaged over all pairs in $\mathcal{P}$, and δ controls the inter-task margin. The rationale of this adjustment is to encourage the representation of each new task to remain closer to its own reference embedding than to those of previously learned tasks. Accordingly, IFA introduces repulsive forces between the current task's global latent representation and previous task references, while maintaining attractive forces toward its own reference. This mechanism promotes inter-task disentanglement and preserves within-task coherence, effectively mitigating representational interference.

If δ is fixed, it cannot adapt to each reference pair. The IFA loss may still take high values even when the current global latent representation is already sufficiently closer to

the current task reference than to the previous task reference (see Supp. for details). To allow δ to adapt to the relative positions of references, we express the margin in terms of the distance between task references:

$$\delta = \alpha \, d\Big(h^{(r)}(T_k),\, h^{(r)}(T_j)\Big) \quad , \tag{5}$$

where α controls the margin. To magnify differences between close task references, instead of using the cosine distance, we propose to use the angular formulation of the distance:

$$d(a, b) = \arccos\left(\frac{a^\top b}{\|a\|_2 \|b\|_2}\right) \tag{6}$$

In this way, our method can adapt to the distance between the task references. Overall, during the lifelong stage, our training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{BC}} + \lambda_{\text{IFA}} \, \mathcal{L}_{\text{IFA}} \, . \tag{7}$$

3.3. Implementation Details

Details for Base Policy. Our policy (see Fig. 2) consists of: vision, language, and state encoders; a FiLM layer that modulates visual and state features via language features; a temporal decoder; and a policy head. The inputs include *agent-view* and *eye-in-hand view* images (see definitions of such modalities in Sec. 4) Specifically, we employ the image and text encoders from CLIP encoders [21] as the vision and language encoders, respectively. A two-layer MLP serves as the state encoder, and a GPT-2 decoder [20] is used as the temporal decoder. The vision and state embeddings are modulated by FiLM layers [18] to obtain task-conditioned representations. The rehearsal buffer $\mathcal{B}$ is populated by randomly sampling trajectories from the current task rollouts $\mathcal{T}_k$, while maintaining a balanced allocation across all previously encountered tasks.

Task Pair Selection for IFA. Task pairs are drawn from the set $\mathcal{P}$ based on their similarity in both the language and agent-view modalities (see modalities definitions in Sec. 4). For each task, we extract the latent features $h^{(l)}$ (language) and $h^{(a)}$ (agent-view) from the current task data as well as from the previously stored samples in the buffer. For any two tasks T_i and T_j, the modality-specific similarity is computed as the mean cosine similarity across all pairs of timesteps:

$$\text{Sim}_e(T_i, T_j) = \frac{1}{N_i N_j} \sum_{t_1=1}^{N_i} \sum_{t_2=1}^{N_j} \frac{h_{t_1}^{(e)}(T_i)^\top h_{t_2}^{(e)}(T_j)}{\|h_{t_1}^{(e)}(T_i)\|_2 \, \|h_{t_2}^{(e)}(T_j)\|_2}. \tag{8}$$

where $e \in \{\text{agent-view}, \text{language}\}$. We compute Sim_a and Sim_l for all task pairs, rank them by similarity, and apply IFA only to pairs that (1) simultaneously fall within the top 50% of most similar pairs in both modalities and (2) include one newly introduced and one previously learned task.

Reference Choice. We take the language embedding $h^{(l)}$ of each task as its reference $h^{(r)}(T_i)$ for IFA, being the task language description informative for the task at hand, and with a stable and fixed representation. We empirically justify our choice in the experimental section.

4. Experimental Analysis

Benchmark Suite and Datasets. We conduct all experiments using the LIBERO lifelong robotic manipulation benchmark [14]. This benchmark provides a diverse set of manipulation tasks that closely resemble human daily activities, such as turning on a stove, moving books, or opening drawers. Each task is specified by a natural language instruction, for example: "*Open the top drawer of the cabinet and put the bowl in it.*"

We follow the setup of prior SOTA, e.g., [23, 26], and adopt three task suites from the LIBERO benchmark: **LIBERO-OBJECT** (10 tasks), **LIBERO-GOAL** (10 tasks), and **LIBERO-50** (50 kitchen tasks selected from LIBERO-100). These suites respectively evaluate the robot's ability to perform diverse actions, and handle complex long-horizon task compositions. All datasets incorporate visual, linguistic, and proprioceptive information, comprising *agent-view* and *eye-in-hand* cameras, textual task instructions, executed actions, and internal robot states, supporting multimodal policy learning. Specifically, *agent-view* provides a visual perspective of the scene, offering spatial and contextual information about the environment, objects, and overall scene layout. *Eye-in-hand* captures fine-grained visual information from a camera mounted on the robot's end-effector. *Language* encodes textual task instructions, guiding the policy by providing semantic goals and context. *Action* refers to the current control commands executed by the robot. Finally, *state* encodes the robot's proprioceptive information, including joint positions, velocities, and gripper status.

During the pretraining stage, six tasks are used for both LIBERO-OBJECT and LIBERO-GOAL ($|\mathcal{T}'| = 6$), and 25 tasks are used for LIBERO-50 ($|\mathcal{T}'| = 25$), with each pretraining task containing 50 demonstrations. In the lifelong learning stage, LIBERO-OBJECT and LIBERO-GOAL consist of four stages, with one new task introduced at each stage, while LIBERO-50 comprises five stages, with five new tasks added per stage to maintain computational feasibility. Each new task in the lifelong stage includes 10 demonstrations.

Evaluation Metrics. We evaluate policy performance using three standard metrics commonly employed in lifelong learning [14, 22, 26]: Forward Transfer (FWT), Negative Backward Transfer (NBT), and Area Under the Curve

Table 1. Comparison of results on the LIBERO benchmarks (mean ± standard deviation). "NA" indicates not available. Best results are shown in bold.

Method	LIBERO-OBJECT			LIBERO-GOAL			LIBERO-50		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
Sequential [14]	62.0±1.0	63.0±2.0	30.0±1.0	55.0±1.0	70.0±1.0	23.0±1.0	32.0±1.0	90.0±2.0	14.0±2.0
ER [2, 14]	56.0±1.0	24.0±1.0	49.0±1.0	53.0±1.0	36.0±1.0	47.0±2.0	35.0±3.0	49.0±1.0	36.0±3.0
BUDS [26, 33]	52.0±2.0	21.0±1.0	47.0±1.0	50.0±1.0	39.0±1.0	42.0±1.0	29±3.0	50.0±4.0	33.0±3.0
LOTUS [26]	74.0±3.0	**11.0±1.0**	65.0±3.0	61.0±3.0	30.0±1.0	56.0±1.0	39±2.0	43.0±1.0	45.0±2.0
ISCIL [10]	71.7±1.9	11.9±5.1	66.3±3.7	70.4±3.6	19.4±4.1	60.5±2.5	47.8±1.3	15.0±4.8	37.7±2.1
M2Distill [23]	75.0±3.0	8.0±5.0	69.0±4.0	71.0±1.0	20.0±3.0	57.0±2.0	NA	NA	NA
TAIL [15]	41.5±2.9	22.5±2.1	38.2±2.5	41.0±1.2	22.7±1.1	37.4±1.3	31.8±3.0	19.3±9.4	20.1±3.3
MLR (Ours)	83.3±2.6	12.3±2.4	77.6±3.0	78.8±2.2	10.0±7.5	74.6±2.7	58.0±4.0	20.6±9.0	54.7±2.4
MLR + IFA (Ours)	**84.6±1.9**	11.4±5.6	**79.4±1.5**	**80.0±2.5**	**6.9±0.9**	**77.2±1.8**	**60.8±2.8**	**8.6±6.2**	**56.1±1.8**

(AUC). All three metrics are computed in terms of success rates [14]. While these metrics are widely used, their computation can slightly vary across studies; therefore, for consistency and fair comparison, we adopt the same definitions and calculation procedures as the studies we compare our method with, e.g., LOTUS [26].

Let M denote the total number of tasks in the lifelong learning sequence, and $r_{i,j}$ denote the agent's success rate on task j after it has learned the first i tasks. FWT measures how well the agent adapts to new tasks and is defined as $\text{FWT} = \frac{1}{M}\sum_{m=1}^{M} r_{m,m}$; higher FWT indicates faster adaptation. NBT measures forgetting on previous tasks, with lower values indicating less forgetting: $\text{NBT}_m = \frac{1}{M-m}\sum_{q=m+1}^{M}(r_{m,m} - r_{q,m})$, and $\text{NBT} = \frac{1}{M-1}\sum_{m=1}^{M-1}\text{NBT}_m$. AUC reflects average performance across all tasks, computed as $\text{AUC} = \frac{1}{M}\sum_{m=1}^{M}\text{AUC}_m$, where $\text{AUC}_m = \frac{1}{M-m+1}\left(r_{m,m} + \sum_{q=m+1}^{M} r_{q,m}\right)$; higher AUC indicates better overall success.

Setup. Our experiments are repeated across three random seeds, with 20 trials per seed–each trial starting from a different configuration of the environment, i.e. different initial state of the gripper. They are conducted on a single NVIDIA A100 GPU, with the pretraining stage and lifelong learning stage both using a batch size of 10 and trained for 100 epochs. We employed the AdamW optimizer [16] with a learning rate initialized to 10^{-4} and a linear scheduler. The weighting coefficient λ_{IFA} is fixed to 0.1. The best angular scaling factor α are 0.3, 0.7, and 0.1 for LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-50, respectively (see Supp. Mat.). We store the multimodal features in the buffer during MRL, with a total size equivalent to five demonstrations per task, following SOTA [26].

4.1. Results

4.1.1. Comparison with SOTA Methods

To assess the capability of our method to learn new tasks while mitigating forgetting of previous ones, we compare its performance against state-of-the-art methods (see Tab. 1). We include several different methods for comparison: Sequential, which fine-tunes each new task in order, using the ResNet–Transformer architecture proposed in [14]; Experience Replay (ER) [2, 14], which stores raw trajectories in a buffer; BUDS, a hierarchical policy-based method [33], which was adapted in [26] to support LIL, LOTUS [26], ISCIL [10] and M2Distill [23]; We also adapt TAIL [15] to the LIL setting, following the protocol of LOTUS [26] (see Supp. Mat. for details).

The proposed MLR already yields substantial improvements over existing SOTA, such as LOTUS [26], ISCIL [10], and M2Distill [23]. In particular, MLR achieves higher FWT and AUC while maintaining competitive NBT values, confirming that our replay strategy alone effectively promotes knowledge retention and forward transfer. When combined with IFA, the performance further improves on every metric. MLR + IFA achieves the highest FWT and AUC and among the lowest NBT values across all benchmarks. For example, on LIBERO-GOAL it improves AUC from 60.5 (ISCIL) to 77.2 while reducing NBT from 19.4 to 6.9, indicating a clear reduction in catastrophic forgetting. On the more challenging LIBERO-50 benchmark, MLR + IFA attains a large margin over all baselines, demonstrating superior scalability to longer and more diverse task sequences. These results confirm that MLR provides an efficient and stable mechanism for continual imitation learning, while the IFA module further refines the replayed representations to enhance both stability and transfer. The combination delivers the best balance between knowledge reuse and plasticity, establishing a new SOTA on all LIBERO benchmarks.

4.1.2. Ablation Study

We conducted a series of experiments to investigate several key aspects of the proposed method. In Tab. 1, we compare our method using latent replay alone (MLR) and with the IFA component included (MLR + IFA). As seen, in-

Table 2. Influence of different modality similarity on the proposed method's performance. "AV", "Lan", "EIH", "Act", and "Sta" denote Agent-view, Language, Eye-in-hand, Action, and State modalities, respectively.

Modality	LIBERO-OBJECT			LIBERO-GOAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
AV	83.8±2.2	**11.1±10.4**	78.9±2.6	77.7±2.6	7.8±2.7	73.9±2.6
Lan	83.3±0.7	13.5±3.1	77.3±1.9	77.1±1.9	12.3±0.8	71.8±2.0
EIH	84.4±0.9	16.1±3.1	77.1±1.1	75.0±3.3	5.6±3.9	72.7±4.2
Act	**84.6±2.6**	20.7±3.6	75.1±1.4	73.8±2.5	**2.5±4.2**	73.2±1.1
Sta	82.5±3.3	11.2±10.6	77.5±2.6	78.3±5.1	8.2±2.4	74.9±5.9
Lan + Act	82.5±3.3	11.2±10.6	77.5±2.6	75.0±3.3	5.6±3.9	72.7±4.2
Lan + EIH	83.3±1.4	16.5±9.0	75.6±5.3	75.0±3.3	5.6±3.9	72.7±4.2
Lan + Sta	**84.6±2.6**	17.4±6.4	76.7±0.6	75.0±3.3	5.6±3.9	72.7±4.2
Lan + AV	**84.6±1.9**	11.4±5.6	**79.4±1.5**	**80.0±2.5**	6.9±0.9	**77.2±1.8**

Table 3. Ablation on task pair selection proportion used when calculating IFA.

	Buffer	FWT↑	NBT↓	AUC↑
LIBERO-OBJECT	33.3%	83.3±2.6	12.3±2.4	77.6±3.0
	50%	**84.6±1.9**	**11.4±5.6**	**79.4±1.5**
	66.6%	83.3±1.4	16.4±9.0	75.8±5.3
LIBERO-GOAL	33.3%	80.0±2.5	**6.9±0.9**	**77.2±1.8**
	50%	80.0±2.5	**6.9±0.9**	**77.2±1.8**
	66.6%	**82.0±0.9**	15.4±16.7	75.6±8.7

Table 4. Effectiveness of choosing different references. "Lan" stands for language. See the text for a description of the "mean global" reference.

	Reference	FWT↑	NBT↓	AUC↑
LIBERO-OBJECT	Mean global	79.6±7.21	**8.3±12.2**	75.7±1.1
	Lan+agent view	**84.6±1.9**	11.4±5.6	**79.4±1.5**
LIBERO-GOAL	Mean global	77.1±3.1	15.7±6.1	70.5±6.0
	Lan+agent view	**80.0±2.5**	**6.9±0.9**	**77.2±1.8**

corporating IFA improves performance across all datasets and metrics. Below, we perform several ablations to examine the impact of different design factors, such as modality pairing, buffer configuration, and loss formulation, on the overall performance. Further results and analyses are included in the Supp. Mat.

Influence of Different Modality Similarity on IFA. Based on the task pair selection for IFA described in Sec. 3.3, we analyze how using an alternative modality e to compute $\text{Sim}_e(T_i, T_j)$ affects the effectiveness of IFA and, consequently, the overall performance of the proposed method. In addition to evaluating each modality individually, we also consider combinations with language, since language serves as the reference in our design. As shown in Tab. 2, the language and agent-view pairing (that is the one our approach leverages) achieves the best overall performance, improving both AUC and FWT while maintaining low forgetting (NBT).

Table 5. Impact of varying the probability of storing features in the buffer, affecting buffer size.

	Probability	Storage	FWT↑	NBT↓	AUC↑
LIBERO-OBJECT	0.50	188MB	84.6±1.9	11.4±5.6	79.4±1.5
	0.20	75.3MB	80.5±7.5	12.5±7.1	77.7±10.0
	0.10	37.6MB	79.9±12.6	13.4±6.1	76.6±14.1
LIBERO-GOAL	0.50	121.2MB	80.0±2.5	6.9±0.9	77.2±1.8
	0.20	60.6MB	78.3±1.9	10.6±8.0	77.4±4.6
	0.10	30.3MB	78.7±1.3	25.2±10.4	66.2±6.8

Effect of Task Pair Selection Proportion. After identifying the language and agent-view modalities as the criteria for IFA as shown in Tab. 2, we further investigate how varying the proportion of selected task pairs influences the performance. The experimental setup is consistent with the previous section, with the only difference being that the independent variable is the selection proportion of task pairs. As shown in Tab. 3, selecting the top 50% of task pairs consistently achieves the highest FWT and AUC while maintaining low NBT, indicating the most stable and effective overall performance.

Effectiveness of Choosing Different References. Different from using the latent feature of language as the reference, i.e., $h^{(r)}(T_k) = h^{(l)}(T_k)$, we also consider the centroid of each task's global latent representations. For previous tasks, we computed the mean of all the global latent representations for each task. During the lifelong learning of the next stage, this means representation serves as the reference for the corresponding previous task. For the new task, we use the mean of its global latent representations accumulated from the start of the current epoch as its reference. All other settings remain unchanged. As shown in Tab. 4, using the latent feature of language as the reference consistently achieves better performance than the averaged global representation.

Impact of Varying Data Storage Ratios. We investigate the impact of varying the buffer size on lifelong learning performance, by setting the probability of storing features in the buffer. In particular, we use either 0.10, 0.20, or 0.50 probability of storing a feature for each new task demonstration. As shown in Tab. 5, retaining fewer representations generally leads to performance degradation, while storing a larger fraction consistently yields more stable results across tasks.

Angle-Based versus Cosine Distance-Based Loss. We further compare the angle-based loss described in Sec. 3.2 with the cosine distance loss. As shown in Fig. 3, the angle-based loss calculation consistently outperforms the cosine distance-based calculation across all metrics. Notably, even the cosine distance–based calculation surpasses all SOTA

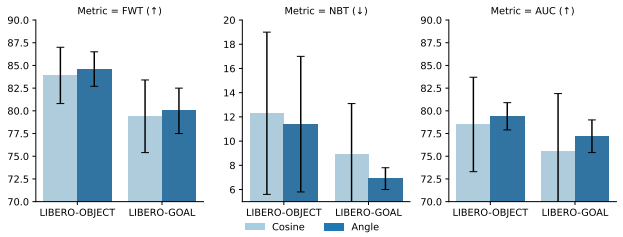
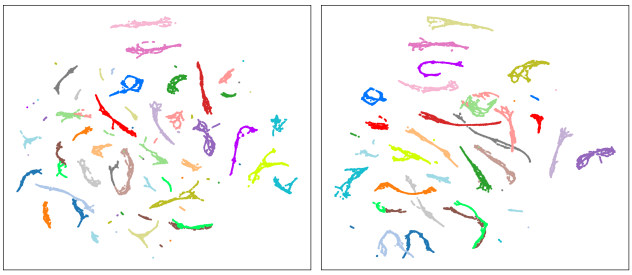


Figure 3. Comparison between cosine distance and angle-based IFA loss calculation.



(a) w/o IFA. (b) w/ IFA.

Figure 4. UMAP visualization of the global latent representation $g(T_k)$ obtained when not enforcing the IFA loss (left) or when enforcing it (right). Each color represents the global latent representations of one task.

Table 6. Average cosine similarity between task 8 embeddings and references from other tasks after lifelong training. "w/o" and "w" denote training without and with IFA, respectively (see main text for details).

	LIBERO-OBJECT			**LIBERO-GOAL**		
	T6	T7	T8	T6	T7	T8
w/o	0.1265	0.1374	0.3590	0.2191	0.2319	0.3652
w	0.1197	0.1218	0.3482	0.1977	0.2345	0.3877

methods and improves upon the MLR-only setting (see Tab. 1). However, its standard deviation is notably higher.

IFA effects in the latent space. To better highlight the advantages brought by the proposed angle-based IFA in the learning dynamics, we compare the global latent representation distributions obtained both training without applying IFA (denoted with "w/o") or applying it (denoted with "w"), with all other settings identical, using UMAP [17] visualization, shown in Fig. 4. Notice that all experimental settings are identical except for the presence of IFA. Without the contribution coming from the IFA loss (Fig. 4-left), embeddings of different tasks are intertwined, showing low inter-task separation and moderate intra-task cohesion, which can lead to catastrophic forgetting as new task features may overwrite old ones. When the IFA component is enforced (Fig. 4-right), embeddings form compact, distinct clusters around their task-specific references, reflecting higher inter-task separation and improved intra-task cohesion. This aligns with the IFA loss, which penalizes embeddings closer to other tasks' references than their own, enforcing repulsion between tasks and attraction to their own reference. The bottom plot, together with the quantitative results presented in this paper, confirms that IFA successfully produces task-aware embeddings, reducing interference and stabilizing lifelong learning by organizing the feature space into distinct, task-specific clusters. Notably, Fig. 4 also shows some task embeddings are already separated without IFA. In this situation, IFA targets such high-similarity cases using an adaptive, angle-based margin, which preserves small distinctions that cosine similarity compresses. Therefore, very similar tasks might still remain highly similar or overlapping.

To further examine IFA's effect on task-level feature separation, we report in Tab. 6 the average cosine similarity between task 8 and references from other tasks after lifelong training. Higher cosine similarity indicates closer alignment in the reference space, which can lead to interference between tasks. On LIBERO-OBJECT, where IFA is only applied between tasks 8 and 7 (i.e., no IFA used for any other task pairs), the similarity between task 8 and task 7 decreases from 0.1374 to 0.1218. Similarly, on LIBERO-GOAL, where IFA is only applied between tasks 8 and 6, the similarity between task 8 and task 6 decreases from 0.2191 to 0.1977. Such consistent reductions indicate that IFA successfully increases inter-task separability as intended.

5. Conclusion

We presented a lifelong imitation learning framework that enables agents to continuously acquire new skills while mitigating catastrophic forgetting. By having our Multimodal Latent Replay and Incremental Feature Adjustment, our method efficiently stores and reuses compact multimodal representations, leveraging frozen pretrained encoders to achieve robust performance even when tasks share similar latent features. Our approach sets a new SOTA on LIBERO benchmarks, improving task transfer and reducing forgetting across diverse multimodal tasks. Looking forward, exploring more complex or longer task sequences, cross-domain and real-world robotic scenarios, and integration with reinforcement learning offers exciting opportunities to further extend the adaptability and generality of our framework, addressing challenges that are broadly difficult in lifelong learning research.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] 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. 1, 2, 6

[3] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023. 2

[4] 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)*, pages 6747–5754. IEEE, 2021. 1, 2

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2

[6] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pages 214–231. Springer, 2024. 2

[7] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017. 1

[8] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 1

[9] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1

[10] 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. 2, 3, 6, 1

[11] Yuheng Lei, Sitong Mao, Shunbo Zhou, Hongyuan Zhang, Xuelong Li, and Ping Luo. Dynamic mixture of progressive parameter-efficient expert library for lifelong robot learning. *arXiv preprint arXiv:2506.05985*, 2025. 1

[12] Yanbang Li, Ziyang Gong, Haoyang Li, Xiaoqi Huang, Haolan Kang, Guangping Bai, and Xianzheng Ma. Robotic visual instruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12155–12165, 2025. 1

[13] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023. 2

[14] 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. 1, 2, 4, 5, 6

[15] 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. *arXiv preprint arXiv:2310.05905*, 2023. 1, 2, 3, 6

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[17] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 8

[18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 5, 1, 2

[19] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3

[20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 5, 1

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2, 5, 1

[22] Natalia Díaz Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. *CoRR*, 2018. 5

[23] Kaushik Roy, Akila Dissanayakc, Brendan Tidd, and Pcyman Moghadam. M2distill: Multi-modal distillation for lifelong imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1429–1435. IEEE, 2025. 1, 2, 3, 5, 6

[24] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33: 13139–13150, 2020. 1

[25] Toshiaki Tsuji, Yasuhiro Kato, Gokhan Solak, Heng Zhang, Tadej Petrič, Francesco Nori, and Arash Ajoudani. A survey on imitation learning for contact-rich tasks in robotics. *arXiv preprint arXiv:2506.13498*, 2025. 1

[26] Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 537–544. IEEE, 2024. 1, 2, 3, 5, 6

[27] Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual

robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3153–3160. IEEE, 2024. 1

[28] Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv e-prints*, pages arXiv–2402, 2024. 2

[29] Yuanqi Yao, Siao Liu, Haoming Song, Delin Qu, Qizhi Chen, Yan Ding, Bin Zhao, Zhigang Wang, Xuelong Li, and Dong Wang. Think small, act big: Primitive prompt learning for lifelong robot manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22573–22583, 2025. 1

[30] Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024. 1

[31] Boyuan Zheng, Sunny Verma, Jianlong Zhou, Ivor W Tsang, and Fang Chen. Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):6322–6337, 2022. 1

[32] Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 57(8):1–35, 2025. 1

[33] 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. 2, 6, 1

Lifelong Imitation Learning with Multimodal Latent Replay and Incremental Adjustment

Supplementary Material

This supplementary material includes extended implementation details of our method, as well as descriptions of the state-of-the-art methods used for comparison in Sec. A. Next, we present additional ablation studies that reflect and justify the design choices of our components and their parameter configurations in Sec. B. Sec. C offers further analyses on IFA and the choice of the task reference embedding, including an explanation of why angle-based computation can be more effective and how the adaptive parameter is designed. Finally, we report the computational efficiency of our method in Sec. D.

A. Extended Implementation Details

A.1. Our Framework

During multi-task pre-training, each task $\mathcal{T}'$ includes $N_j = 50$ expert demonstrations. In the lifelong learning stage, each new task $\mathcal{T}_k$ provides $N_k = 10$ demonstrations, following the setup used in prior state-of-the-art (SOTA) methods [23, 26]. Each rollout in the LIBERO suites consists of 70–300 time steps.

Our architecture follows the general design introduced in prior work [15], where each action is predicted from the previous $L = 8$ observations. Specifically, we employ CLIP-base [21] as both the vision encoder and the language encoder, each implemented as a 12-layer Transformer. A 6-layer GPT-2 decoder [20] serves as the temporal decoder, processing the visual and state tokens from the last eight steps and outputting the parameters of a 5-component Gaussian Mixture Model (GMM) over continuous actions. The use of a GMM policy head is a common practice in the LIBERO benchmarks [14]. Task instructions are injected via FiLM-based [18] conditioning of the visual and state representations within the modulation network. During pre-training, we train only LoRA adapters (with rank-8) in the CLIP backbones, together with the temporal decoder and the GMM policy head. In the lifelong learning stage, only the temporal decoder and policy head are updated for each new task, as depicted in Figure 2 in the main paper.

A.2. State of the Art

For the methods reported in Tab. 1 of the main paper, Sequential [14], Experience Replay (ER) [2, 14], BUDS [26, 33], and LOTUS [26] all use the same architectural setup and training configuration as LOTUS. Specifically, LOTUS employs a skill-discovery module to predict subgoal actions, which are then composed into a full action sequence to accomplish the task, and the numerical results for these methods are taken directly from the original LOTUS paper [26]. As for ISCIL [10], each skill is associated with its own adapter, and during evaluation the method retrieves the adapter corresponding to the skill most similar to the current input. For ISCIL, we use the implementation from [11], which has been adapted for the LIBERO setting. In this way, the same training setup as other methods is followed, ensuring fair comparisons.

We exploited an in-house implementation of TAIL [15] (note that the official code for TAIL [15] is not available). We verified that our implementation has the same number of parameters as reported in the TAIL paper, and therefore use it for all comparisons with confidence. As the official implementation is unavailable, we ensured architectural and training consistency with the original paper to maintain fair comparisons. We will make the corresponding code publicly available upon acceptance of our paper to foster future research in this direction.

TAIL [15] is a task-ID–based method that trains a separate adapter for each task to adapt the shared backbone to the corresponding data distribution. At test time, TAIL requires the task ID to select the correct adapter. To make TAIL compatible with the task-ID–agnostic LIL setting considered in the main paper, we first train it following the original procedure, without any modification. During evaluation, since multiple task-specific adapters are available at a given stage of the lifelong phase, we compute the success rate of each task as the average performance across all adapters available up to that point. This averaging simulates task-ID agnostic inference, where the correct adapter cannot be selected, and follows the same evaluation strategy adopted in LOTUS [26] when adapting task-incremental methods to LIL. All reported metrics use this averaged success rate, ensuring a fair comparison with LIL approaches that employ a single shared policy.

B. Additional Ablation Studies

B.1. Influence of the Angular Scaling Parameter α in IFA

We conducted an ablation study to evaluate the impact of different values of the angular scaling factor α in the adaptive IFA formulation (Tab. 7). The best-performing values of α are 0.3, 0.7, and 0.1 for LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-50, respectively.

The adaptive margin δ is defined as proportional to

Table 7. Ablation on different angular scaling factor α in the adaptive IFA.

Dataset	α	FWT↑	NBT↓	AUC↑
LIBERO-OBJECT	0.10	81.8±3.2	12.4±3.6	77.3±1.4
	0.30	**84.6±1.9**	11.4±5.6	**79.4±1.5**
	0.50	82.1±0.7	18.3±10.3	73.2±4.1
	0.70	81.7±0.7	**8.4±3.3**	77.3±2.4
LIBERO-GOAL	0.10	**80.8±4.0**	12.0±11.3	75.4±8.8
	0.30	81.3±3.3	15.2±7.3	74.7±6.7
	0.50	79.2±1.9	12.8±2.8	73.1±2.3
	0.70	80.0±2.5	**6.9±0.9**	**77.2±1.8**
LIBERO-50	0.10	**60.8±2.8**	8.6±6.2	**56.1±1.8**
	0.30	53.6±4.0	**7.6±10.0**	52.0±1.4
	0.50	58.5±3.6	8.9±4.9	53.2±3.4
	0.70	60.5±2.2	10.6±1.1	54.2±2.3

Table 8. Comparison between LoRA adapters with different ranks (R) and full fine-tuning (FFT).

	LIBERO-OBJECT			LIBERO-GOAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
R=8	57.5±0.2	7.9±1.0	52.8±5.9	59.3±0.9	11.5±3.3	54.1±0.7
R=16	48.9±7.1	**-5.6±0.4**	50.4±6.9	56.3±5.3	**4.4±18.9**	53.6±2.4
R=32	53.8±0.2	-0.2±6.7	54.2±2.3	63.8±7.1	7.5±13.0	60.0±1.5
FFT	**84.6±1.9**	11.4±5.6	**79.4±1.5**	**80.0±2.5**	6.9±0.9	**77.2±1.8**

the angular distance between the reference embeddings of the current task and the most similar previous task: $\delta = \alpha \arccos\left(\frac{a^\top b}{|a|_2|b|_2}\right)$. Intuitively, this ensures that tasks with larger semantic differences are more strongly separated in the latent space, while similar tasks remain closer, preserving within-task coherence.

For LIBERO-50, the largest and most diverse benchmark, smaller α (0.1) gives the best overall performance, likely reflecting the denser arrangement of task reference embeddings—larger margins could over-penalize nearby tasks. However, performance at α = 0.7 remains comparable, indicating that the adaptive IFA is relatively robust to the choice of α even in large, densely packed task suites. This suggests that α can be tuned to balance inter-task separation and latent stability without critically affecting performance.

B.2. Adapters versus Full Fine-tuning.

During the lifelong learning stage, our default configuration fine-tunes all parameters of the temporal decoder. For comparison, we also evaluate a parameter-efficient variant based on Low-Rank Adaptation (LoRA) [5], where the temporal decoder is frozen and LoRA modules are inserted into the attention projection layers (query/key/value and output projections) and the MLP feed-forward layers. We experiment with different LoRA ranks, and the results are summarized in Tab. 8.

These results reveal a substantial performance gap: full fine-tuning significantly outperforms the LoRA-based adaptation across all datasets and metrics. This indicates that, in our setting, the temporal decoder requires sufficient capacity to capture complex temporal dynamics and integrate information coming from the MLR and IFA losses, which parameter-efficient approaches alone cannot provide. Furthermore, LoRA introduces additional hyperparameters that require grid search, making it less practical for large-scale lifelong learning scenarios.

Table 9. Effect of using FiLM layers during lifelong learning.

	LIBERO-OBJECT			LIBERO-GOAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
w/o FiLM	43.8±3.5	**8.8±7.7**	41.6±0.2	57.6±17.7	**0.83±7.5**	56.3±2.1
w FiLM	**84.6±1.9**	11.4±5.6	**79.4±1.5**	**80.0±2.5**	6.9±0.9	**77.2±1.8**

B.3. Influence of Using FiLM Layers.

In our model, the outputs of the vision and state encoders are modulated by FiLM layers [18] to obtain task-conditioned representations. To empirically justify their contribution, we ablate the model by removing FiLM layers during the lifelong learning stage (see Tab. 9).

As seen, incorporating FiLM layers significantly improves FWT and AUC compared to the variant without FiLM. Although the no-FiLM variant shows slightly lower NBT, this is due to its overall weaker learning performance rather than genuine resistance to forgetting. These results indicate that explicit feature-wise modulation is crucial for effective adaptation: by dynamically scaling and shifting intermediate activations, FiLM allows the network to adjust its internal representations efficiently to the conditioning signal, improving forward transfer and overall learning effectiveness.

C. Extended Analysis

C.1. Selection of a stable task reference

As described in the main paper, our proposed Incremental Feature Adjustment (IFA) is a regularization technique designed to counter representation drift across tasks during Lifelong Learning (LIL). IFA achieves this by ensuring that the global latent representation ($g_t(T_k)$), which is the output of the temporal decoder for the current task T_k, remains closer to its own task reference ($h^{(r)}(T_k)$) than to the references of previously learned tasks ($h^{(r)}(T_j)$). This introduces repulsive forces between the current task's global latent representation and previous task references, promoting inter-task disentanglement and preserving within-task coherence.

The efficacy of IFA relies critically on selecting a stable

Table 10. Task-reference comparison on **LIBERO-OBJECT** and **LIBERO-GOAL**. See text for the explanation of "Mean Global" and "Global". "Language", "AgentView", "Eye-in-hand", and "State" use the corresponding modality-specific latent feature as the task reference. The smallest cosine distance is highlighted in **bold**, and the second smallest is underlined.

	LIBERO-OBJECT				LIBERO-GOAL			
Task Reference	**Task 6**	**Task 7**	**Task 8**	**Task 9**	**Task 6**	**Task 7**	**Task 8**	**Task 9**
Mean global	**0.565**	**0.555**	**0.549**	**0.454**	**0.485**	**0.354**	**0.501**	**0.437**
Global	0.811	0.802	0.797	0.703	0.735	0.583	0.752	0.684
Language	0.668	0.701	0.656	0.677	0.591	0.610	0.629	0.680
AgentView	0.947	0.951	0.973	0.994	0.976	0.957	0.980	1.051
Eye-in-hand	1.008	0.974	1.005	1.005	1.018	1.037	0.987	1.042
State	1.068	1.012	1.024	1.006	1.033	1.072	1.005	1.036

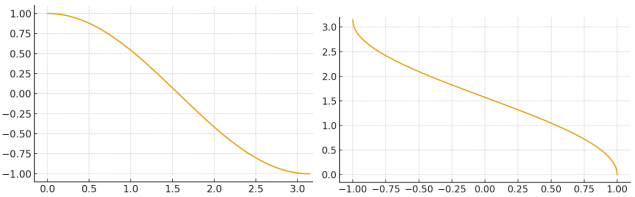
Figure 5. UMAP visualization of the global latent representations $g(T_k)$ obtained when not enforcing the IFA loss, each color represents the global latent representations of one task.

and representative task reference ($h^{(r)}$) for each task. The motivation for this selection can be seen from the visualization in Fig. 5: we observe that the global latent representations of a given task naturally cluster together in the embedding space. This clustering indicates that each task can be summarized by a single, representative point that reflects the overall location of its features.

To justify our choice of the language latent feature as the task reference ($h^{(r)}$), we compared it against other viable candidates. We computed the average cosine distance between these candidates and the global latent representations of their corresponding tasks (on LIBERO-OBJECT and LIBERO-GOAL, Tasks 6–9), evaluating both their representativeness and their stability during incremental learning. The task reference candidates evaluated were the mean latent feature derived from different modalities (first column of Tab. 10): Mean global, the mean of all global latent representations for a task; Global, a measure derived by averaging the distances obtained when treating each individual global latent representation as a separate, unstable candidate reference; Modality-Specific Latent Features, the mean latent feature of a specific modality, such as language, agent view, eye-in-hand, or state (see Section 4). For example, when using the agent-view modality, we take the mean agent-view latent features of the task as the task reference and compute its average cosine distance to all global latent representations of that task (see Tab. 10). This data confirmed that using the mean of all global latent representations (the Global Latent Feature candidate) yields the smallest cosine distance, proving it is the most statistically representative point. Despite being the most representative, the mean global reference is fundamentally unstable. During the ongoing process of Lifelong Incremental Learning (LIL), the dynamic changes in model parameters cause the latent representations, and consequently their mean, to change over time. This drifting reference hinders the ability of IFA to consistently regulate representation drift. Crucially, the language latent feature produced the second-smallest distances. Furthermore, the language-based task reference (which is identical to its mean since the language modality provides a single, fixed embedding per task description) remains stable throughout training. It even outperformed the unstable "Global" baseline. Given that a stable anchor is essential for IFA to effectively prevent representation drift across tasks, we adopt the language latent feature as our definitive task reference ($h^{(r)}$), prioritizing its stability for reliable regularization over the slightly higher representativeness of the unstable mean global feature.

C.2. Extended details for MLR and IFA.

MLR. In the lifelong learning stage, we utilize the proposed Multimodal Latent Replay (MLR). Instead of storing raw sensory data, a compact buffer $\mathcal{B}$ maintains the multimodal latent representations ($\mathbf{H}$), which are concatenated from the frozen encoders, alongside the associated action (a) for training. In subsequent tasks, the stored $\mathbf{H}$ from previous tasks is fed into the temporal decoder for replay.



(a) $\cos(\theta)$ on $[0, \pi]$. (b) $\arccos(x)$ on $[-1, 1]$.

Figure 6. Illustration of the cosine function and its inverse.

IFA. To prevent interference (catastrophic forgetting) between the current task (T_k) and the previously learned tasks (T_j), we introduce the IFA. IFA's core mechanism encourages the global latent representation of the current task ($g_t(T_k)$) to remain closer to its own stable reference ($h^{(r)}(T_k)$) than to the references of previously learned tasks ($h^{(r)}(T_j)$).The IFA loss is explicitly defined to penalize violations of this distance constraint:

$$\mathcal{L}_{\text{IFA}} = \frac{1}{|\mathcal{P}|} \sum_{\substack{(j,k)\in\mathcal{P} \\ j<k}} \max\Big(0,\, d\big(g_t(T_k), h^{(r)}(T_k)\big) - d\big(g_t(T_k), h^{(r)}(T_j)\big) + \delta\Big), \tag{9}$$

The margin δ adapts to the relative positions of the task references themselves:

$$\delta = \alpha\, d\Big(h^{(r)}(T_k),\, h^{(r)}(T_j)\Big) \quad , \tag{10}$$

Angle-based distance. The IFA loss is explicitly driven by the difference in distance between the global latent representation $g_t(T_k)$ and the task references: the positive reference $h^{(r)}(T_k)$ (for the current task T_k) and the negative references $h^{(r)}(T_j)$ (for the previous tasks T_j). Therefore, we compared how the standard cosine similarity and its angular counterpart (distance $d(\cdot,\cdot) = \arccos(\cdot)$) behave when used inside this crucial distance difference.

Fig. 6 illustrates this comparison. The left panel plots $\cos(\theta)$ as a function of the angle $\theta \in [0, \pi]$ between two unit vectors, while the right panel shows $\arccos(x)$ as a function of $x \in [-1, 1]$. Cosine (left) quickly saturates in the high-similarity regime: when two representations are already very close, even noticeable angular changes produce almost no change in $\cos\theta$, making the resulting distance difference extremely small and hard to distinguish. In contrast, the angular distance $\arccos(x)$ (right) expands this high-similarity region; small changes in x near 1 translate into clearly separable angular differences, providing more stable resolution across similarity levels and allowing IFA to better capture small deviations between task-level representations. Outside this range, $\arccos(x)$ behaves similarly to cosine similarity—differences between representations remain distinguishable.

Table 11. Runtime and FLOPs comparison. See text for the explanation of Forward Time, S-FLOPs, and T-FLOPs. All experiments are conducted on **LIBERO-GOAL**. P means the probability of storing features in the buffer, which is ablated in Tab. 5. RR stands for raw replay, which stores past samples and replays them during training. Note that BASE stands for running our method without MLR or IFA. It is included solely to quantify the computational cost of the base architecture.

Method	P	Forward Time (ms)	S-FLOPs	T-FLOPs
BASE	-	75.5 ± 2.43	3.63×10^{11}	4.36×10^{17}
BASE + RR	0.5	75.8 ± 2.80	3.63×10^{11}	12.96×10^{17}
BASE + MLR	0.5	75.5 ± 2.43	3.63×10^{11}	7.62×10^{17}
BASE + MLR + IFA	0.5	75.8 ± 2.80	3.63×10^{11}	7.62×10^{17}
BASE + MLR + IFA	0.2	75.8 ± 2.43	3.63×10^{11}	5.66×10^{17}
BASE + MLR + IFA	0.1	75.8 ± 2.43	3.63×10^{11}	5.01×10^{17}

Adaptive δ. The choice of δ as a fixed hyperparameter would incur in a risk of choosing an overly large margin. Specifically, if δ is set larger than the distance between the two task references, i.e., $\delta > d\big(h^{(r)}(T_k), h^{(r)}(T_j)\big)$, the IFA loss will always enforce a penalty. This is guaranteed by the triangle inequality property of spherical geometry, which states that the difference between the distances to the two references is bounded by the distance between the references themselves:

$$d\big(g_t(T_k), h^{(r)}(T_k)\big) - d\big(g_t(T_k), h^{(r)}(T_j)\big) \le d\big(h^{(r)}(T_k), h^{(r)}(T_j)\big),$$

If δ exceeds this maximum possible difference, the loss term inside the $\max(\cdot)$ will always be positive. This would enforce an unnecessary repulsive force, even when the current global latent representation $g_t(T_k)$ is already sufficiently closer to its own reference $h^{(r)}(T_k)$, potentially destabilizing learning.

To avoid this unnecessary penalty and allow the margin to adapt to the distance between references, we define δ dynamically:

$$\delta = \alpha\, d\big(h^{(r)}(T_k), h^{(r)}(T_j)\big), \quad \alpha \in (0, 1).$$

This guarantees that the margin is always bounded by the inter-reference distance, and scales proportionally to how far apart the tasks are in the angular space.

D. Computational efficiency

We report the computational efficiency using several primary metrics related to policy execution in Tab. 11. Forward Time refers to the time required to produce one action (i.e., a single forward pass). We computed forward time

using a NVIDIA A100 GPU, by first running a warming up stage of 10 inferences (to avoid transient effects from initialization and caching), and then computing the average and standard deviation on 100 subsequent inferences.

The S-FLOPs correspond to the total number of Floating-Point Operations (FLOPs) required for a single forward pass of the policy to produce one action. This metric reflects model inference cost and is independent of data or training procedure. For completeness, we also report T-FLOPs, the total training FLOPs, computed as per-forward FLOPs multiplied by the total number of forward pass, backward pass, and gradient update during training (i.e., FLOPs × epochs × training sampless × 3).

As shown in the result table, both MLR and IFA have a minimal overhead on per-inference efficiency, indicating that their additional computations introduce negligible runtime cost. In terms of total training FLOPs, adding MLR (P = 0.5) increases the computational cost by roughly $0.75\times$ on **LIBERO-GOAL** due to replay, and the less the buffer is, the less the T-FLOPs. But this increase remains reasonable and acceptable given the resulting performance gains compared to the performance in Tab. 5.