LOTUS: Continual Imitation Learning for Robot Manipulation Through Unsupervised Skill Discovery

Weikang Wan[1,2], Yifeng Zhu*[1], Rutav Shah*[1], Yuke Zhu[1]

***Abstract*— We introduce LOTUS, a continual imitation learning algorithm that empowers a physical robot to continuously and efficiently learn to solve new manipulation tasks throughout its lifespan. The core idea behind LOTUS is constructing an ever-growing skill library from a sequence of new tasks with a small number of human demonstrations. LOTUS starts with a continual skill discovery process using an open-vocabulary vision model, which extracts skills as recurring patterns presented in unsegmented demonstrations. Continual skill discovery updates existing skills to avoid catastrophic forgetting of previous tasks and adds new skills to solve novel tasks. LOTUS trains a meta-controller that flexibly composes various skills to tackle vision-based manipulation tasks in the lifelong learning process. Our comprehensive experiments show that LOTUS outperforms state-of-the-art baselines by over 11% in success rate, showing its superior knowledge transfer ability compared to prior methods. More results and videos can be found on the project website: https://ut-austin-rpl.github.io/Lotus/.**

I. Introduction

Deploying robots in the open world necessitates continual learning in ever-changing environments. Imagine you bring home a new appliance — your future home robot is unlikely to have seen it before and must quickly learn to operate it. Such a scenario pinpoints the importance of lifelong learning capabilities [48], with which a robot can continually learn and adapt its behaviors over time. Lifelong robot learning is particularly challenging as it involves constant adaptation under distribution shifts throughout a robot's lifespan.

A plethora of literature has investigated lifelong learning with monolithic neural networks [13, 16, 19, 56, 60]. As these monolithic models have constant capacities, they often fall short in complex domains such as vision-based manipulation [26], where the ever-growing set of tasks would eventually incur insurmountable computational burdens. Alternatively, the computational burden can be greatly reduced by exploiting the compositional structures of underlying tasks. Such structures can be identified as the recurring segments among the trajectories, corresponding to reusable skills across different tasks [63]. Another body of work has harnessed the recurring patterns in task structures and extracted *skills* for knowledge transfer [40, 42, 47, 55]. These works have demonstrated that skills can be composed by a hierarchical model to scaffold new behaviors more efficiently than their monolithic counterparts. Nonetheless, they either assume a fixed set of skills, thus limiting the range of behaviors they can express, or demand a prohibitively high sample complexity for physical robots.

[1]The University of Texas at Austin, [2]Peking University. * Equal contributions.

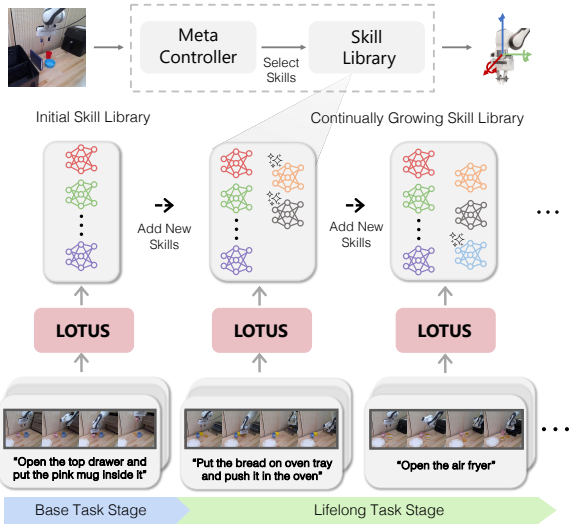

Fig. 1: **Method Overview.** LOTUS is a continual imitation learning algorithm through unsupervised skill discovery. LOTUS starts from the base task stage, where it builds an initial library of sensorimotor skills. In the subsequent lifelong task stage, it continuously discovers new skills from a stream of incoming tasks and adds them to its skill library. A high-level meta-controller composes skills from the library to solve new manipulation tasks. We mark the newly acquired skills in the library with ✨.

Recently, imitation learning [7, 52, 61, 62] has shown great promise in tackling robot manipulation tasks. These algorithms offer a data-efficient framework for acquiring sensorimotor skills from a small set of human demonstrations, often collected directly on real robots. Hierarchical imitation learning methods [25, 29, 59] further harness temporally extended skills to address long-horizon manipulation tasks that require prolonged interactions and diverse behaviors. Nonetheless, most of these works have focused on learning skills from a single task or a fixed set of tasks known *a priori*. These assumptions hinder their effectiveness in lifelong settings, where the sequence of new tasks results in non-stationary data distribution.

This work examines the problem of *continual imitation learning* for real-world robot manipulation. We aim to develop a practical algorithm that learns over a sequence of new tasks a physical robot may encounter. Our algorithm builds up a continually growing skill library by adding new skills and updating old ones. Our goal is to efficiently learn a policy that leverages skills extracted from past experiences to achieve better performance on new tasks (*forward transfer*) while retaining its competitive performance on previously learned tasks (*backward transfer*).

To this end, we introduce LOTUS (**L**ifel**O**ng knowledge **T**ransfer **U**sing **S**kills). The crux of our method, as illustrated in Figure 1, is to build an ever-growing library of sensorimotor skills from a stream of new tasks. Every skill in this library is modeled as a goal-conditioned visuomotor policy that operates on raw images. To harness the skill library, LOTUS trains a meta-controller to invoke one skill by index, specifying its behavior by generating subgoals at each time step. We train LOTUS in the continual imitation learning fashion [13, 26], where each new task comes with a small number of human demonstrations collected through teleoperation. LOTUS starts from a set of base tasks, acquiring its initial library of skills. Then, in the subsequent lifelong task stage, we introduce an incremental skill clustering process to partition temporal segments of new task demonstrations to determine whether 1) to add a new skill or 2) to update an existing skill with new training data. Consequently, we obtain an ever-increasing skill library for solving previous and new tasks in the lifelong learning process.

We systematically evaluate LOTUS in continual imitation learning for vision-based manipulation, both in simulation and on a real robot. LOTUS reports over 11% higher average success rates than the state-of-the-art baselines during the lifelong learning stages. Our results show the efficacy of using skills to achieve better *forward* and *backward transfer* in lifelong learning settings. Qualitatively, we find that LOTUS tends to discover new skills to interact with novel object concepts or generate new motions.

In summary, our contributions are three-fold: 1) We introduce a hierarchical learning algorithm that continually discovers sensorimotor skills from a stream of new tasks with human demonstrations; 2) We show that LOTUS transfers skills to solve new manipulation tasks more effectively than baselines; 3) We systematically analyze LOTUS's performances and successfully deploy it to physical hardware.

II. Related Work

Lifelong Learning for Decision Making. Lifelong learning aims to develop generalist agents that adapt to new tasks in ever-changing environments [21, 34, 38, 47, 53]. Prior works have trained monolithic policies [5, 16, 27, 57, 58], but this methodology is shown ineffective in knowledge transfer for robot manipulation [26]. Alternatively, another line of research attempts to leverage skills through compositional modeling of lifelong learning tasks [3, 6, 35, 57]. They attempt to enable more efficient knowledge transfer compared to the monolithic policy counterparts. These methods, primarily based on hierarchical reinforcement learning, induce high sample complexity. They fail to scale to complex domains such as vision-based manipulation. Unlike prior works, LOTUS uses skills in a hierarchical imitation learning framework with experience replay to enable sample-efficient learning while effectively transferring (backward and forward) knowledge in vision-based manipulation domains.

Skill Discovery in Robot Manipulation. Skill discovery studies how a robot identifies recurring segments of sensorimotor experiences, often termed skills. Many studies have tackled skill discovery through self-exploration with hierarchical reinforcement [1, 12, 14, 22, 24, 51], or with information-theoretic bottlenecks [11, 17, 44]. However, these works demand high sample complexity and often rely on ground-truth physical states, hindering them from applying to real robot hardware. another line of works discover skills from human demonstrations [11, 17, 44]. More recent works have successfully discovered skills from demonstrations purely based on raw sensory cues, creating a collection of closed-loop visuomotor policies to tackle long-horizon manipulations. [8, 46, 63]. However, these works assume fixed state-action distributions in multitask settings, preventing them from tackling continually changing situations during the robot's lifespan. LOTUS differs from prior work in that it discovers skills from demonstrations collected in a sequence of tasks. This approach addresses the challenge of skill discovery in a dynamic environment where the data distribution is constantly changing.

Hierarchical Imitation Learning. LOTUS uses hierarchical imitation learning with temporal abstractions to acquire sensorimotor skills for complex tasks [30, 45]. Specifically, we employ hierarchical behavior cloning, a promising technique in robot manipulation [15, 31, 32, 49, 52, 63]. These methods factorize a policy into a two-level hierarchy, where a high-level policy predicts subgoals and low-level parameterized skill policies generate motor commands. However, existing methods only work with a fixed set of skills in multitask settings, limiting their use in lifelong learning where the number of skills grows over time. In contrast, LOTUS uses hierarchical behavioral cloning with Experience Replay [5], enabling the learning of varying numbers of skill policies.

III. Background

In this section, we introduce the formulation of vision-based robot manipulation and continual imitation learning. These two formulations are essential to LOTUS.

Vision-Based Manipulation. We formulate a vision-based robot manipulation task as a finite-horizon Markov Decision Process: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, H, \mu_0, R)$. Here, $\mathcal{S}$ is the space of the robot's raw sensory data, including RGB images and proprioception. $\mathcal{A}$ is the space of the robot's motor commands. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is transition dynamics. H is the maximal horizon for each episode of tasks. μ_0 is the initial state distribution. $R(s, a, s')$ is the reward function. We consider a sparse-reward setting, where the reward function is defined by a goal predicate $g : \mathcal{S} \mapsto \{0, 1\}$. Each task $T^m \equiv (\mu_0^m, g^m)$ is defined by the initial state distribution μ_0^m and the goal predicate g^m. The robot's objective is to learn a policy π that maximizes the expected return: $\max_\pi J(\pi) = \mathbb{E}_{s_t, a_t \sim \pi, \mu_0}[\sum_{t=1}^{H} g(s_t)]$.

Continual Imitation Learning. We consider a continual imitation learning setting [26] as mentioned in Section I, which allows sample efficient policy learning over tasks with sparse rewards. In this setting, a robot sequentially trains a policy π using imitation learning over M tasks $\{T^m\}_{m=1}^{M}$.

A robot encounters the tasks in sequence $T^{1:m_1}$, $T^{m_1:m_2}$, ..., $T^{m_{C-1}:m_C}$, where $1 < m_c < M (1 \leq c < C)$ and $m_C = M$. We break the lifelong learning process into two stages, a *base task stage* for learning a multitask policy over $T^{1:m_1}$, and a *lifelong task stage* where a robot sequentially learns all the other tasks $T^{m_1:M}$. For clarity of presentation, the rest of the descriptions in Section III and IV assume one task is learned at every step c during the lifelong task stage.

For each task T^m, a robot is provided with a small dataset of N demonstrations for T^m, denoted as $D^m = \{\tau_i^m\}_{i=1}^N$ and a language description l^m. The policy is conditioned on the observations and the task specification, i.e., $\pi(\cdot|s; l^m)$. The demonstrations are collected through expert teleoperation with each trajectory consisting of state action pairs $\tau_i^m = \{(s_t, a_t)\}_{t=1}^{t_m}$ where $t_m < H$. π is trained with a behavioral cloning loss [2] that clones the demonstrated actions. The lifelong learning setting assumes the robot loses full access to $\{D^p : p < m\}$ when learning T^m [54]. Therefore, naive multitask learning methods result in catastrophic forgetting in lifelong learning due to memory bottleneck [21]. Our goal is to learn a sample-efficient policy that quickly learns on new tasks (*forward transfer*) while retaining its success rates on previously learned tasks (*backward transfer*).

Evaluation Metrics. We use three standard metrics to evaluate policy performance in lifelong learning [10, 26, 28]: FWT (forward transfer), NBT (negative backward transfer), and AUC (area under the success rate curve). All three metrics are calculated in terms of success rates [26], where a higher FWT suggests quicker adaptation to new tasks, a lower NBT indicates better performance on past tasks, and a higher AUC means better average success rates across all tasks evaluated. Denote $r_{i,j}$ as the agent's success rates on task j when it has just learned over previous i tasks. These three metrics are defined as follows: $\text{FWT} = \sum_{m\in[M]} \frac{r_{m,m}}{M}$, $\text{NBT} = \sum_{m\in[M]} \frac{\text{NBT}_m}{M}$, $\text{NBT}_m = \frac{1}{M-m} \sum_{q=m+1}^{M} (r_{m,m} - r_{q,m})$, and $\text{AUC} = \sum_{m\in[M]} \frac{\text{AUC}_m}{M}$, $\text{AUC}_m = \frac{1}{M-m+1}(r_{m,m} + \sum_{q=m+1}^{M} r_{q,m})$.

IV. Method

We present LOTUS, a hierarchical imitation learning method for robot manipulation in a lifelong learning setting. Key to LOTUS is building an ever-increasing library of skill policies through continual skill discovery. LOTUS segments demonstrations into temporal segments based on DINOv2 [37] features. We leverage these features to form semantically consistent clusters of temporal segments in the presence of non-stationary data distribution during lifelong learning. Throughout the lifelong learning process, LOTUS continually adds new skills to facilitate new task learning while refining existing ones without catastrophic forgetting using a skill clustering method adapted from unsupervised incremental clustering [41]. Moreover, a meta-controller is trained with hierarchical imitation learning to harness the skill library. Figure 2 shows the overview of LOTUS.

Hierarchical Imitation Learning With Continual Skill Discovery. LOTUS aims to learn a hierarchical policy that sample-efficiently adapts to new tasks continually during the lifelong learning process while retaining performance on previous tasks. The hierarchical policy learning considers the policy π with a two-level hierarchy: the low level is a set of skills from the continually growing skill library $\{\pi_k^L\}_{k=1}^K$ and the high level is a meta-controller π^H that invokes the skills. At lifelong learning stage c, a policy is factorized as: $\pi(a_t|s_t, l) = \pi^H(\omega_t, k_t|s_t, l)\pi_{k_t}^L(a_t|s_t, \omega_t)$, where $l \in \{l^m\}_{m=1}^{M_c}$, $k_t \in [K_c]$, ω_t is the subgoal parameter. K_c is the maximal number of skills discovered until step c. In continual imitation learning [21], the hierarchical policy is trained over a sequence of tasks that come with demonstrations. Since demonstrations from previous tasks are not fully available in a lifelong setting, LOTUS uses Experience Replay (ER) to learn policies with a memory buffer B_c that saves some exemplar data for each task introduced before step c.

Key to training π in LOTUS is to maintain a growing skill library $\{\pi_k^L\}_{k=1}^K$. This goal entails continually clustering demonstrations of new tasks into skill partitions, which we term *continual skill discovery*. Specifically, at a lifelong learning step c, demonstrations are split into maximally K_c partitions $\{\tilde{D}_k\}_{k=1}^{K_c}$. Each partition $\tilde{D}_k$ is used to fine-tune an existing skill policy k when $k \leq K_{(c-1)}$ or train a new skill policy if $K_{(c-1)} < k \leq K_c$. If a partition k is empty at step c, we do not update π_k^L. Note that the base task stage is equivalent to continual skill discovery in multitask learning.

A. Continual Skill Discovery With Open-World Perception

For continually discovering new skill policies $\{\pi_k^L\}_{k=1}^K$, it is crucial to split demonstrations D^m of a task T^m into partitions $\{\tilde{D}_k\}_{k=1}^{K_c}$ where $\tilde{D}_k$ consists of training data for skill k. The key to curating partitions for training skill policies is to identify the recurring temporal segments in the demonstration of new tasks. LOTUS first uses a bottom-up hierarchical clustering approach [63] to temporally segment demonstrations and incrementally cluster temporal segments into partitions.

Temporal Segmentation with Open-World Vision Model. To recognize recurring patterns for obtaining the partitions, LOTUS first needs to identify the coherent temporal segments from demonstrations based on scene similarity [63]. We apply hierarchical clustering on each demonstration based on agglomerative clustering [23], which breaks a demonstration into a sequence of disjoint temporal segments in a bottom-up manner. The essence of the hierarchical clustering step is to merge temporally adjacent segments that are most semantically similar until a task hierarchy is formed, from which the temporal segments can be decided [63].

The primary challenge of applying hierarchical clustering in the lifelong setting is consistently identifying the semantic similarity between temporally adjacent segments in a non-stationary data distribution of lifelong learning. We use DINOv2 [37], an open-world vision model that can output consistent semantic features of open-world images [61], allowing LOTUS to reliably measure the semantic similarity between temporally adjacent segments. Specifically, to incorporate temporal information, we aggregate DINOv2

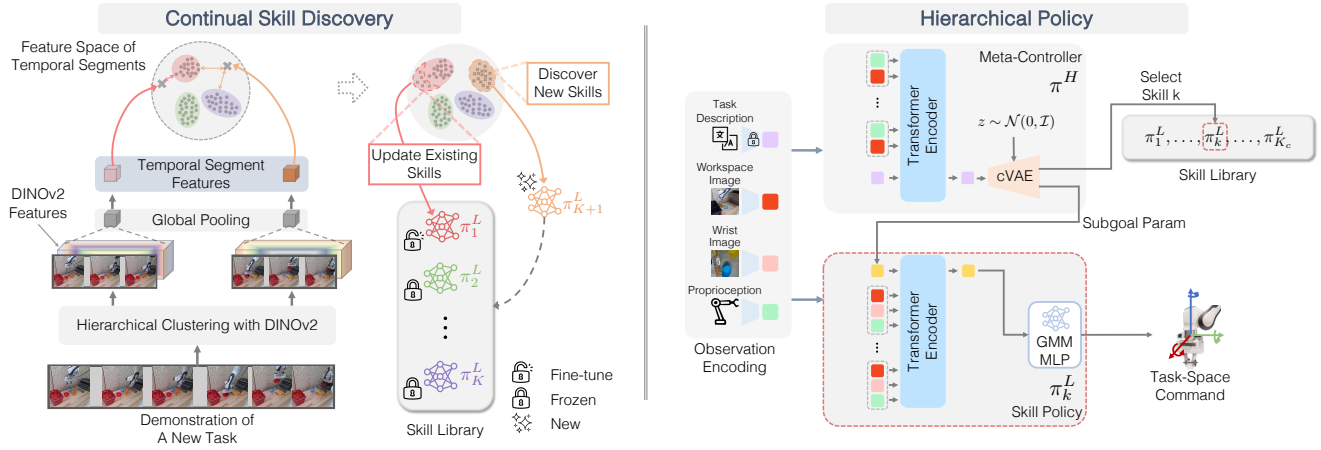

Fig. 2: LOTUS consists of two processes: continual skill discovery with open-world perception and hierarchical policy learning with the skill library. For continual skill discovery, we obtain temporal segments from demonstrations using hierarchical clustering with DINOv2 features and incrementally cluster the temporal segments into partitions to either update existing skills or learn new skills. For the hierarchical policy, a meta-controller π^H selects a skill by predicting an index k and specifies the subgoals for the selected skill π_k^L to achieve. Note that because the input to a transformer is permutation invariant, we also add sinusoidal positional encoding to input tokens to inform transformers of the temporal order of input tokens [50]. We omit this information in the figure for clarity.

features of all frames in the segment using a global pooling operation. Then, the semantic similarity between consecutive temporal segments is quantified using the cosine similarity scores between pooling features.

Incremental Clustering for Skill Partitions. The identified temporal segments from demonstrations pave the way for subsequent steps of identifying recurring patterns among tasks and clustering them into partition data used to train skill policies. We develop an incremental skill clustering method that segments demonstrations into temporal segments using unsupervised skill discovery [63] and clusters the temporal segments into either existing or new partitions as follows: In the *base task stage*, LOTUS uses spectral clustering [41] to first partition the demonstrations into K_1 skills. LOTUS determines the value of K_1 using the Silhouette method [41], which quantifies the score of how well data points match with their clusters on a scale of -1 to 1. We sweep through the integer values of K_1 to find the one with the highest Silhouette value. LOTUS then continually groups new temporal segments into increasing partitions in the subsequent *lifelong learning stages*, where it either adds a segment to an existing partition to help backward transfer or creates a new partition which is used to train a new skill to learn new behaviors that facilitate forward transfer.

At the lifelong learning step c, LOTUS calculates the Silhouette value between new temporal segments and the previous K_{c-1} partitions to determine the new partitions. Segments with Silhouette values above a threshold are grouped with the partition having the highest value, indicating high similarity between the temporal segment and existing skill. In contrast, segments with values below the threshold are assigned to a new partition. This process is repeated serially for all segments. Our preliminary results indicate that the clustering is robust and tolerates a range of threshold values. After partitioning new segments into partitions, LOTUS clusters demonstrations D^m into partitions $\{\tilde{D}_k\}_{k=1}^{K_c}$, each corresponding to an existing or new skill.

B. Hierarchical Imitation Learning With Experience Replay

LOTUS uses the partitions ($\{\tilde{D}_k\}_{k=1}^{K_c}$) obtained from continual skill discovery to train the skill library $\{\pi_k^L\}_{k=1}^{K}$, from which the meta-controller π_H can invoke individual skills. In a lifelong learning setting, with no full access to demonstrations from previous tasks, LOTUS uses Experience Replay (ER) [5] to learn the policies for its effectiveness in knowledge transfer [26]. Concretely, LOTUS trains the policy using behavior cloning with a dataset that consists of demonstrations D^m of the new task at step c and data stored in the memory buffer B_c. After learning, LOTUS saves a subset of demonstration trajectories $D'^m \subset D^m$ into the buffer: $B_{c+1} = B_c \cup D'^m$.

In the following parts, we describe skill policy learning, where LOTUS keeps track of skill partitions in the memory buffer. Then, we describe the design of the meta-controller that invokes skills from the continually growing library.

Skill Policy Learning. The partitions $\{\tilde{D}_k\}_{k=1}^{K_c}$ from continual skill discovery provide the training datasets for each skill policy. Along with the partitions, to retain previous knowledge while finetuning the skill policies, LOTUS leverages exemplar data from the memory buffer B_c that also retains partition information of the saved demonstrations. The saved partition for a skill k at a learning step c is denoted as $B_{k,c}$ in buffer B_c. Every existing skills π_k^L is fine-tuned using $\tilde{D}_k \cup B_{k,c}$, while newly-created skills are directly trained on $\tilde{D}_k$. After the policy finishes training on task T^m, the memory buffer updates the information of skill partitions with a subset of demonstrations $\tilde{D}_k' \subset \tilde{D}_k$ for partition k: $B_{k,c+1} = B_{k,c} \cup \tilde{D}_k'$.

LOTUS models each skill as a goal-conditioned visuomotor policy, which allows the meta-controller to specify which subgoal for the skill to achieve. LOTUS encodes

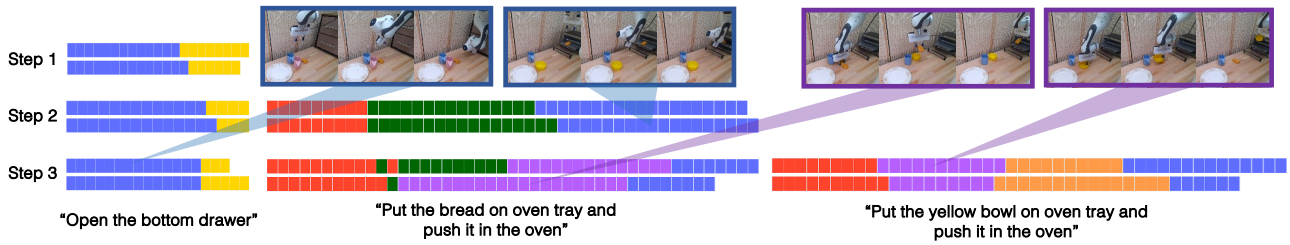

Fig. 3: LOTUS continually discovers skills from real-world tasks in the robot's lifespan of learning (each color represents a skill). The skill (in blue) used for "reaching the bottom drawer" is also used for "pushing the oven tray inside" due to similar motion in action space. It shows the forward transfer of skills enabled by LOTUS. The skill (in purple) discovered in Step 3 of "putting the yellow bowl on the oven tray" is used for a previous task, "putting the bread on the oven tray," demonstrating the backward transfer of learned skills in our method.

the look-ahead images within temporal segments using a subgoal encoder into the subgoal embedding ω_t and is jointly trained with skill policies. Representing the subgoals in latent space makes the meta-controller computationally tractable to predict ω_t during inference. The history of input images from the workspace and wrist cameras are encoded with ResNet-18 encoders [18] before passing it through the transformer encoder [50, 62] along with subgoal embedding ω_t. For computing the actions, the output token from the transformer encoder corresponding to ω_t is then passed into a Gaussian Mixture Model (GMM) neural network [4, 33] to model the multi-modal action distribution in demonstrations.

Skill Composition With Meta-Controller. To harness the learned library of skill policies, we use a meta-controller π^H to compose the skills. π^H is designed for two purposes: selects a skill k, and specifies subgoals (ω_t) for the selected skill to achieve. Given a task description l^m, π^H decides which skill and subgoal to achieve by considering the current task progress. To understand the task progress, π^H perceives the current layouts of objects and the robot's states captured by workspace images and robot proprioception, respectively. As the true state of the task is partially observable, the meta-controller takes a history of past observations as inputs and allows temporal modeling of the underlying states by using a transformer. The observation inputs are converted to tokens with ResNet-18 encoders [18], whereas l^m is encoded into a language token by a pretrained language model, Bert [9], before passing it to the transformer encoder. To capture the multi-modal distribution of skill indices and subgoals underlying human demonstrations, LOTUS trains a conditional Variational Auto-Encoder (cVAE) by minimizing an ELBO loss over demonstrations [20]. The training supervision for the meta-controller comes from continual skill discovery and skill policy learning: 1) The labels of skill indices k_t from the clustering step; 2) The subgoal embeddings ω_t obtained from encoding look-ahead images of temporal segments using subgoal encoder. It is important to note that the subgoal encoder is jointly trained with skill policy learning.

The meta-controller must handle a variable number of skills due to the ever-growing nature of the skill library. To address this challenge, we design the meta-controller with an output head that predicts a sufficiently large number of skills, noted K_{max}. Then, we introduce a binary mask whose first K_c entries are set to 1 at a lifelong learning step c. The mask limits the skill index prediction to the existing set of skills, and when new skills are added, the meta-controller can predict more skills based on modified masking. Note that the mask does not change back when the policy is evaluated on tasks prior to step c, so that π_H can transfer new skills to previously learned tasks. Concretely, the meta-controller first predicts logits $z \in \mathcal{R}^{K_{max}}$, then applies masking to z which returns modified logits denoted as z' where $z'_k = z_k$ if $k \leq K_c$; otherwise, $z'_k = -\infty$. The probability for a given skill k is computed as $\text{Softmax}(z'_k) = \frac{e^{z'_k}}{\sum_{j=1}^{K\max} e^{z'_j}}$. Applying masking directly to the output probabilities from logits z would require reweighting the probability during lifelong learning, giving rise to numerical instability. Our masking design mitigates this issue.

V. Experiments

We design the experiments to answer the following questions: 1) Does hierarchical policy design improve knowledge transfer in the lifelong setting? 2) Do the newly discovered skills facilitate knowledge transfer? 3) Is the hierarchical design of LOTUS more sample-efficient than baselines that do not use skills? 4) How does the choice of large vision models affect knowledge transfer? 5) Is LOTUS practical for real-robot deployment?

A. Experimental Setup

Simulation Experiments. We conduct evaluations in simulation using the task suites from the lifelong robot learning benchmark, LIBERO [26]. We select three suites, namely LIBERO-OBJECT (10 tasks), LIBERO-GOAL (10 tasks), and LIBERO-50 (50 kitchen tasks from LIBERO-100). The benchmarks evaluate the robot's ability to understand different object concepts, execute different motions, and achieve both, respectively. Experiments over the three task suites emulate various complexities of lifelong learning, which comprehensively evaluate the performance of LOTUS and other baselines. For the base task stage, we choose $m_1 = 6$ tasks for LIBERO-OBJECT and LIBERO-GOAL and $m_1 = 25$ for LIBERO-50. In all the task suites, each base task has $N = 50$ demonstrations. LIBERO-OBJECT and LIBERO-GOAL have $C = 4$ lifelong learning steps and 1 task per steps, whereas LIBERO-50 has $C = 5$ steps and 5 tasks per step considering the computation burden. Following the convention of ER [5] algorithm design in LIBERO, each task introduced in the lifelong learning stage has $N = 10$

Tasks	Evaluation Setting	SEQUENTIAL [26]	ER [5]	BUDS [63]	LOTUS-ft	LOTUS
LIBERO-OBJECT	FWT (↑)	62.0 ± 0.0	56.0 ± 1.0	52.0 ± 2.0	68.0 ± 4.0	**74.0** ± 3.0
	NBT (↓)	63.0 ± 2.0	24.0 ± 0.0	21.0 ± 1.0	60.0 ± 1.0	**11.0** ± 1.0
	AUC (↑)	30.0 ± 0.0	49.0 ± 1.0	47.0 ± 1.0	34.0 ± 2.0	**65.0** ± 3.0
LIBERO-GOAL	FWT (↑)	55.0 ± 0.0	53.0 ± 1.0	50.0 ± 1.0	56.0 ± 0.0	**61.0** ± 3.0
	NBT (↓)	70.0 ± 1.0	36.0 ± 1.0	39.0 ± 1.0	73.0 ± 1.0	**30.0** ± 1.0
	AUC (↑)	23.0 ± 0.0	47.0 ± 2.0	42.0 ± 1.0	26.0 ± 1.0	**56.0** ± 1.0
LIBERO-50	FWT (↑)	32.0 ± 1.0	35.0 ± 3.0	29.0 ± 3.0	32.0 ± 2.0	**39.0** ± 2.0
	NBT (↓)	90.0 ± 2.0	49.0 ± 1.0	50.0 ± 4.0	87.0 ± 2.0	**43.0** ± 1.0
	AUC (↑)	14.0 ± 2.0	36.0 ± 3.0	33.0 ± 3.0	16.0 ± 1.0	**45.0** ± 2.0

TABLE I: Main Experiments. Results are averaged over three seeds and we report the mean and standard error. All metrics are computed in terms of success rates (%).

Metrics	CLIP	R3M	DINOv2
FWT (↑)	12.0 ±1.0	57.0 ±4.0	61.0 ±3.0
NBT (↓)	43.0 ±2.0	33.0 ±2.0	30.0 ±1.0
AUC (↑)	29.0 ±2.0	52.0 ±3.0	56.0 ±1.0

TABLE II: Ablation study on using different large vision models for continual skill discovery.

demonstrations. LOTUS stores 5 demonstrations for each task in subsequent lifelong learning steps.

Real Robot Tasks. We evaluate LOTUS on a real robot manipulation task suite, MUTEX [43] (50 tasks). They include a variety of tasks, such as "open the air fryer and put the bowl with hot dogs in it." We choose $m_1 = 25$ for the base task stage, and $C = 5$ steps containing 5 tasks per step in the lifelong task stage. In real-world experiments, each base task includes $N = 30$ demonstrations, while tasks introduced during lifelong learning have $N = 10$. LOTUS retains 5 demonstrations per task in subsequent lifelong learning steps. Additional details are provided on our project website.

B. Quantitative Results

For all simulation experiments, we compare LOTUS against multiple baselines in each task suite for 20 trials per task, repeated for three random seeds (Table I). We evaluate models with FWT, NBT, and AUC, defined in Section III. We compare our method with the following baselines:

- SEQUENTIAL: Naively fine-tuning the new tasks in sequence using the ResNet-Transformer architecture from LIBERO [26].
- ER [5]: Experience Replay baseline using the ResNet-Transformer without the inductive bias of skills.
- BUDS [63]: A hierarchical policy baseline that learns policies from multitask skill discovery. We adopt BUDS to lifelong learning by re-training it every lifelong step, using the same policy architecture as LOTUS.
- LOTUS-ft: LOTUS variant which only fine-tunes the new task demonstrations without ER.

Table I provides a comprehensive evaluation of LOTUS and the baselines in simulation. It answers question (1) by showing that LOTUS consistently outperforms the best baseline, ER, across all three metrics. Additionally, while ER yields worse FWT than its fine-tuning counterpart, SEQUENTIAL (a consistent finding shown in the prior work [26]), LOTUS yields better FWT than its fine-tuning counterpart, LOTUS-ft. This result shows that the inductive bias of skills in policy architectures is important for a memory-based lifelong learning algorithm to transfer previous knowledge to new tasks effectively.

Ablative Studies. We use LIBERO-GOAL for conducting all ablations. We answer the question (2) by comparing LOTUS with its variant without adding new skills, which decreases by 13.0, −3.0, and 7.0 in FWT, NBT, and AUC, respectively. The performance declines in all metrics when no new skills are introduced, highlighting the importance of expanding the skill library for LOTUS to achieve better knowledge transfer. To address the question (3), we incrementally increase the demonstrations per task for training ER. The AUC are 0.47 (10 demos), 0.53 (15 demos), 0.53 (20 demos), and 0.57 (25 demos), respectively. ER only surpasses LOTUS when using 25 demonstrations, implying significantly better data efficiency of LOTUS compared to other baselines.

To answer question (4), we compare our DINOv2-based method with other large vision models that are pretrained on Internet-scale datasets and human activity datasets, namely CLIP [39] and R3M [36]. The result in Table II shows that our choice of DINOv2 is the best for continual skill discovery. At the same time, R3M is also significantly better than CLIP at skill discovery, even though R3M performs worse than DINOv2. Note that our open-world vision model choice is not limited to DINOv2 and can be replaced by superior models in the future.

Real Robot Results. We compare LOTUS with the best baseline ER on MUTEX tasks. Our evaluation shows that LOTUS achieved 50 in FWT (+11%), 21 in NBT (+ 2%), and 56 in AUC (+9%) in comparison to ER. The performance over the three metrics shows the efficacy of LOTUS policies on real robot hardware, answering the question (5). This result also shows that our choice of DINOv2 features is general across both simulated and real-world images. Additionally, we visualize the skill compositions during real robot evaluation in Figure 3, showing that LOTUS not only transfers previous skills to new tasks but also achieves promising results of transferring new skills to previous tasks.

VI. CONCLUSION

We introduce LOTUS, a continual imitation learning method for building vision-based manipulation policies with skills. LOTUS tackles continual skill discovery by using an open-world vision model to extract recurring patterns in unsegmented demonstrations and an incremental skill clustering method that clusters demonstrations into an increasing number of skills. LOTUS continually updates the skill library to avoid catastrophic forgetting of previous tasks and adds new skills to exhibit novel behaviors. LOTUS uses hierarchical imitation learning with experience replay to train both the skill library and the meta-controller that adaptively composes various skills.

Currently, LOTUS requires expert human demonstrations through teleoperation, which can be costly. For future work,

we intend to look into discovering skills from human videos. Moreover, LOTUS still requires storing demonstrations of previously learned tasks in the experience replay to ensure effective forward and backward transfers. It would incur large memory burdens if the number of tasks increases to thousands. For future work, we plan to investigate compressing data from prior tasks to improve the memory efficiency of our algorithm.

ACKNOWLEDGMENTS

This work was completed during Weikang Wan's visit to the Robot Perception and Learning (RPL) Lab at UT Austin. RPL research has been partially supported by the National Science Foundation (CNS-1955523, FRR-2145283), the Office of Naval Research (N00014-22-1-2204), UT Good Systems, and the Machine Learning Laboratory.

REFERENCES

[1] A. Bagaria and G. Konidaris, "Option discovery using deep skill chaining," in *ICLR*, 2019.

[2] M. Bain and C. Sammut, "A framework for behavioural cloning.," in *Machine Intelligence 15*, 1995, pp. 103–129.

[3] E. Ben-Iwhiwhu, S. Nath, P. K. Pilly, S. Kolouri, and A. Soltoggio, "Lifelong reinforcement learning with modulating masks," *arXiv preprint arXiv:2212.11110*, 2022.

[4] C. M. Bishop, "Mixture density networks," 1994.

[5] A. Chaudhry *et al.*, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[6] L. Chen, S. Jayanthi, R. R. Paleja, D. Martin, V. Zakharov, and M. Gombolay, "Fast lifelong adaptive inverse reinforcement learning from demonstrations," in *Conference on Robot Learning*, PMLR, 2023, pp. 2083–2094.

[7] C. Chi *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[8] V. Chu *et al.*, "Real-time multisensory affordance-based control for adaptive object manipulation," in *ICRA*, IEEE, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] N. Dıéaz-Rodrıéguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: New metrics for continual learning," *arXiv preprint arXiv:1810.13166*, 2018.

[11] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *arXiv:1802.06070*, 2018.

[12] R. Fox, S. Krishnan, I. Stoica, and K. Goldberg, "Multi-level discovery of deep options," *arXiv:1703.08294*, 2017.

[13] C. Gao, H. Gao, S. Guo, T. Zhang, and F. Chen, "Cril: Continual robot imitation learning via generative and prediction model," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 6747–5754.

[14] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," *arXiv:1611.07507*, 2016.

[15] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," in *CoRL*, 2020, pp. 1025–1037.

[16] S. Haldar and L. Pinto, "Polytask: Learning unified policies through behavior distillation," *arXiv preprint arXiv:2310.08573*, 2023.

[17] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller, "Learning an embedding space for transferable robot skills," in *ICLR*, 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[19] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.

[21] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[22] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *NIPS*, vol. 22, pp. 1015–1023, 2009.

[23] S. Krishnan *et al.*, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *IJRR*, 2017.

[24] V. C. Kumar, S. Ha, and C. K. Liu, "Expanding motor skills using relay networks," in *CoRL*, 2018.

[25] H. Le, N. Jiang, A. Agarwal, M. Dudıék, Y. Yue, and H. Daumé, "Hierarchical imitation and reinforcement learning," in *ICML*, 2018, pp. 2917–2926.

[26] B. Liu *et al.*, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

[27] S. Liu *et al.*, "Continual vision-based reinforcement learning with group symmetries," in *7th Annual Conference on Robot Learning*, 2023.

[28] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[29] J. Luo *et al.*, "Multi-stage cable routing through hierarchical imitation learning," *arXiv preprint arXiv:2307.08927*, 2023.

[30] C. Lynch *et al.*, "Learning latent plans from play," in *CoRL*, 2020.

[31] A. Mandlekar *et al.*, "Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," in *ICRA*, IEEE, 2020.

[32] A. Mandlekar *et al.*, "Learning to generalize across long-horizon tasks from human demonstrations," in *RSS*, 2020.

[33] A. Mandlekar *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[34] J. A. Mendez, L. P. Kaelbling, and T. Lozano-Pérez, "Embodied lifelong learning for task and motion planning," *arXiv preprint arXiv:2307.06870*, 2023.

[35] J. A. Mendez, H. van Seijen, and E. Eaton, "Modular lifelong reinforcement learning via neural composition," *arXiv preprint arXiv:2207.00429*, 2022.

[36] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[37] M. Oquab *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[38] S. Powers, A. Gupta, and C. Paxton, "Evaluating continual learning on a home robot," *arXiv preprint arXiv:2306.02413*, 2023.

[39] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[40] A. Raghavan, J. Hostetler, I. Sur, A. Rahman, and A. Divakaran, "Lifelong learning using eigentasks: Task separation, skill acquisition, and selective transfer," *arXiv preprint arXiv:2007.06918*, 2020.

[41] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[42] N. M. Shafiullah and L. Pinto, "One after another: Learning incremental skills for a changing world," *arXiv preprint arXiv:2203.11176*, 2022.

[43] R. Shah, R. Martín-Martín, and Y. Zhu, "MUTEX: Learning unified policies from multimodal task specifications," in *7th Annual Conference on Robot Learning*, 2023.

[44] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," in *ICLR*, 2019.

[45] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner, "Taco: Learning task decomposition via temporal alignment for control," in *ICML*, 2018, pp. 4654–4663.

[46] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Learning manipulation graphs from demonstrations using multimodal sensory signals," in *ICRA*, IEEE, 2018.

[47] C. Tessler, S. Givony, T. Zahavy, D. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in minecraft," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[48] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and autonomous systems*, vol. 15, no. 1-2, pp. 25–46, 1995.

[49] A. Tung *et al.*, "Learning multi-arm manipulation through collaborative teleoperation," in *ICRA*, 2021.

[50] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[51] A. S. Vezhnevets *et al.*, "Feudal networks for hierarchical reinforcement learning," in *ICML*, 2017.

[52] C. Wang *et al.*, "Mimicplay: Long-horizon imitation learning by watching human play," *arXiv preprint arXiv:2302.12422*, 2023.

[53] G. Wang *et al.*, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[54] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *arXiv preprint arXiv:2302.00487*, 2023.

[55] B. Wu, J. K. Gupta, and M. Kochenderfer, "Model primitives for hierarchical lifelong reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 34, pp. 1–38, 2020.

[56] A. Xie and C. Finn, "Lifelong robotic reinforcement learning by retaining experiences," *ArXiv*, vol. abs/2109.09180, 2021.

[57] A. Xie and C. Finn, "Lifelong robotic reinforcement learning by retaining experiences," in *Conference on Lifelong Learning Agents*, PMLR, 2022, pp. 838–855.

[58] A. Xie, J. Harrison, and C. Finn, "Deep reinforcement learning amidst lifelong non-stationarity," *arXiv preprint arXiv:2006.10701*, 2020.

[59] T. Yu, P. Abbeel, S. Levine, and C. Finn, "One-shot hierarchical imitation learning of compound visuomotor tasks," *arXiv:1810.11043*, 2018.

[60] W. Zhou *et al.*, "Forgetting and imbalance in robot lifelong learning with off-policy data," in *Conference on Lifelong Learning Agents*, PMLR, 2022, pp. 294–309.

[61] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu, "Learning generalizable manipulation policies with object-centric 3d representations," in *7th Annual Conference on Robot Learning*, 2023.

[62] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *arXiv preprint arXiv:2210.11339*, 2022.

[63] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4126–4133, 2022.