Temporal Logic Guided Affordance Learning for Generalizable Dexterous Manipulation

Hao Zhang, Hao Wang, Tangyu Qian, and Zhen Kan

2024 7th International Symposium on Autonomous Systems (ISAS) | 979-8-3503-6317-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISAS61044.2024.10552562

Abstract— **Reinforcement learning has shown great potential to address long-horizon tasks. However, most existing work lacks the ability to reason about functional parts of objects and extract the task semantics to facilitate robotic learning, making it difficult to achieve category-level generalization even on seen objects. To address these issues, we develop a temporal logic guided affordance learning framework for generalizable dexterous manipulations (TALD). In particular, TALD is equipped with affordance learning for predicting actionable information and LTL representation for understanding the task semantics and improving learning efficiency. We design a contact predictor from the affordance learning perspective to improve the generalization performance, which predicts the max-affordance point observation based on the 3D point clouds to guide the agent to manipulate the functional part of objects. And we exploit the LTL progression to construct a task-driven labeled POMDP to address the challenge of a non-Markovian reward function, and design a task module to extract the LTL representation by Transformer Encoder to improve the sampling efficiency and facilitate the robotic learning. We validate the proposed method in four articulated manipulation scenarios. The generalization performance corresponding to success rates and visualization effects show the effectiveness of TALD. Our project is available at: https://sites.google.com/view/tald-0257/.**

I. Introduction

One of the ultimate goals of robotic learning is to let the robot itself understand where the critical parts of an object are and how they can be manipulated efficiently. In order to achieve such human-level intelligence, the capability of reasoning the functional areas of manipulated parts with affordance and understanding the semantics of task semantics is essential. With the development of learning algorithms, reinforcement learning (RL) has shown its potential to address long-horizon tasks, which models the dynamics of interaction as a Markov decision process (MDP) and concentrates on learning an optimal policy from the update by exploiting the transitions sampled from MDP [1]. While RL-based methods have enabled robots to perform tasks ranging from simple to complex (e.g., MuZero [2] and Go-Explore [3]), a significant and yet difficult topic is how robots can determine the optimal operating point for object manipulation and effectively extend this capability to unseen objects. Specifically, there are three primary issues: 1) In contrast to most RL methods that rely on state information fed back from the environment to allow the agent to learn object manipulation, how can the robot effectively generalize to manipulate different objects based on vision information? 2) When performing manipulating on an object with different functional parts, how to improve the contact position from the view of affordance learning? 3) Since there are often similar task instructions when manipulating a class of objects, how the robot can incorporate the task semantics to further improve the performance of their generalization capabilities?

H. Zhang, Hao Wang, Tangyu Qian, and Z. Kan (Corresponding Author) are with the Department of Automation at the University of Science and Technology of China, Hefei, Anhui, China, 230026.

The idea of affordance learning was originally proposed in [4] mainly to suggest possible ways for agents to interact with objects, which has been proven effective in improving the performance of operations from different perspectives. An affordance representation model based on self-supervised learning is proposed in [5] to calculate the distance between the robot and the manipulated object as well as to determine the optimal interaction point for global to local object grasping. The work [6] considers affordance learning from an interaction perspective and proposes a learning framework to estimate per-pixel affordance map for achieving articulated tasks. The work [7] further extends [6] by considering the ability of two agents manipulating in concert to accomplish challenging tasks. The work of [8] proposes an end-to-end manipulation learning framework by exploiting RL and affordance learning, which collects contact information to optimize the affordance model and drives the agent to get better performance based on the predicted interaction point. Although there have been advancements, it remains challenging how traditional approaches that rely on affordance learning can be effectively combined with task semantics to direct the agent in manipulating tasks that involve a series of sub-goals that must be logically accomplished.

Due to the rich expressivity and capability, linear temporal logic (LTL) is capable of describing a wide range of complex tasks composed of logically organized sub-tasks [9]–[11]. Learning algorithms are often utilized to tackle complex tasks involving specialized logic by converting the LTL specification into an automaton. For instance, parameterized action reinforcement learning is exploited with LTL representations to make the robot manipulate diverse long-horizon tasks [12]. Learning-based safety manipulation subject to the finite-state predicate automaton (FSPA) in the presence of complex tasks is investigated in [13]. The method of composing the LDGBA and the MDP to an embedded product MDP (EP-MDP) is designed in [14] to solve the difficulty of sparse rewards setting. However, the above algorithms are usually dedicated to solving a specific single task, and are hard to have good generalization performance in similar task scenarios. In [15], a method, namely meta

Q-learning of multi-task, is proposed to generalize the learned model to a new set of tasks that decomposed by LTL Progression. However, this framework is difficult to extend to zero-shot generalization situations. An approach extracting the semantics of LTL based on graph neural networks is proposed in [16] to achieve task-oriented policies and generalize to new instructions. However, it uses radar information to allow the robot to make similar decisions in unseen scenarios, whose sensors are difficult to extend for manipulation tasks. The work [8] exploits 3D point clouds to predict contact positions based on affordance maps and generalizes over different kinds of manipulation tasks. However, it ignores the enhancement of understanding task semantics for robotic manipulations.

To bridge this gap, we consider encoding LTL instructions via Transformer to guide the agent to understand the task semantics to offer better affordance and further improve its generalization ability in similar scenarios.

We summarize the main contributions of this work as follows:

1) We proposed a novel framework for generalizable dexterous manipulation over four articulated object manipulations, namely TALD, which not only guides the agent's manipulation with the max-affordance point observation, but also exploits the task semantics to improve learning efficiency and generalization through category-level evaluation.

2) We design a contact predictor from the affordance learning perspective to improve the generalization performance, which predicts the max-affordance point observation based on the 3D point clouds to guide the agent to manipulate the functional part of objects.

3) We exploit the LTL progression to construct a task-driven labeled POMDP to address the challenge of a non-Markovian reward function, and design a task module to extract the LTL representation by Transformer Encoder to improve the sampling efficiency and facilitate the robotic learning.

4) We validate the proposed method in four articulated manipulation scenarios. The generalization performance corresponding to success rates and visualization effects show the effectiveness of TALD.

II. Preliminaries

A. Co-Safe Linear Temporal Logic

Co-safe LTL (sc-LTL) is a subclass of LTL that can be satisfied by finite-horizon state trajectories [17]. Since sc-LTL is well-suited for describing robotic tasks (e.g., approach the bucket, grasp the handle, and then lift it up to the table), this work will concentrate on sc-LTL. An sc-LTL formula is constructed by using a collection of atomic propositions Π that can be true or false, standard Boolean operators like $\wedge$ (conjunction), $\vee$ (disjunction), and $\neg$ (negation), as well as temporal operators like $\bigcirc$ (next), $\Diamond$ (eventually), and $\cup$ (until). The semantics of an sc-LTL formula are interpreted over a word $\boldsymbol{\sigma} = \sigma_0\sigma_1...\sigma_n$, which is a finite sequence with $\sigma_i \in 2^{\Pi}$, $i = 0, \ldots, n$, where 2^{Π} represents the power set of Π. Denote by $\langle \boldsymbol{\sigma}, i \rangle \vDash \varphi$ if the sc-LTL formula φ holds from position i of $\boldsymbol{\sigma}$. [10] provides more comprehensive explanations and illustrative examples.

B. Labeled POMDP and Visual Reinforcement Learning

When considering an sc-LTL instruction φ in visual RL, a labeled POMDP $\mathcal{M}_e$ can be constructed to model the dynamics between the agent and the environment as $\mathcal{M}_e = (S, T, A, p_e, \Pi, L, R, \gamma, \mu, \Omega, O)$, where S is the state space, $T \subseteq S$ shows a set of terminal states, A represents the action space, $p_e(s'|s, a)$ is the transition probability from $s \in S$ to $s' \in S$ under action $a \in A$, Π is a collection of atomic propositions meaning the properties corresponding to the states, $L : S \to 2^{\Pi}$ denotes the labeling function, $R : S \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1]$ is the discount factor, μ is the initial state distribution. The observation space of an environment is denoted as Ω. The observation function $O : S \to \Omega$ translates an environment state s to its corresponding observation in Ω. And a deterministic policy π_e is used to interact the environment over $\mathcal{M}_e$ by outputting the action a and get the reward by $r_t = R(s_t)$.

The reward function is usually considered to be Markovian, meaning the reward at s_{t+1} only depends on the transition from s_t to s_{t+1}. However, when a robot attempts a long-horizon task, it is usually only rewarded when the task is completed, i.e., $\boldsymbol{\sigma} \vDash \varphi$. Since the word $\boldsymbol{\sigma} = \sigma_0\sigma_1...\sigma_t$ is interpreted from the state trajectory $s_0s_1...s_t$ by the labeling function L, it produces a non-Markovian reward function

$$R(s_0s_1...s_t) = \begin{cases} r_{env} + r_\varphi, & \text{if } \boldsymbol{\sigma} \vDash \varphi \\ r_{env} - r_\varphi, & \text{if } \boldsymbol{\sigma} \vDash \neg\varphi \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $\sigma_t = L(s_t)$, r_{env} is the environmental reward and r_φ corresponds to the task rewards when the instruction φ is satisfied with $\boldsymbol{\sigma}$.

III. Problem Formulation

In order to further explain the motivation of the proposed method for learning dexterous skills with temporal logic and affordance learning, we will illustrate an example that will be considered throughout this work.

Example 1. Consider a dexterous manipulation [18] shown in Fig. 1(a), in which the robot needs to approach the toilet and open the lid O_{lid}. The set of propositions Π is {toilet_approahed, lid_grasped, lid_opened}. Using above propositions in Π, an example sc-LTL formula is $\varphi_{\mathsf{toilet}} = \Diamond(\mathsf{toilet\_approahed} \wedge \Diamond(\mathsf{lid\_grasped} \wedge \Diamond \mathsf{lid\_opened}))$, which requires the robot to sequentially approach, grasp, and open the lid of the toilet. The possible functional part and predicted manipulated point are highlighted in red and green, respectively, as shown in Fig. 1(b).

In this work, we are interested in exploiting affordance learning to achieve category-level generalization based on the point cloud and absorbing task semantics encoded by Transformer to further improve the performance. By segmenting the 3D point cloud of the scene and extracting

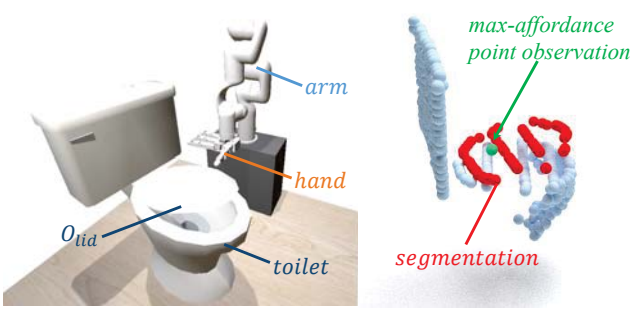

Fig. 1: We illustrate an example that the robot implements some dexterous manipulation (e.g., φ_{toilet}) by manipulating the object in the environment.

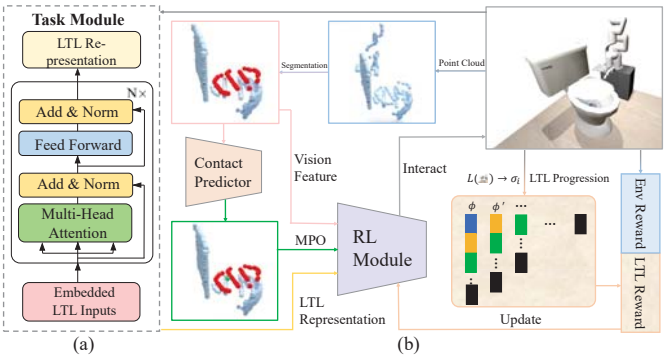

Fig. 2: The framework of TALD that exploits the contact predictor and LTL representation to improve the learning efficiency and ensures generalization performance. a) Task Module: The outline of encoding the LTL instructions φ to the LTL representation φ_θ by Transformer Encoder. (b) The main method in TALD to implement category-level generalization over a wide range of objects which first extracts the vision feature from 3D point clouds, then predicts the MPO by the contact predictor and interacts the environment with the guidance of the MPO and LTL representation.

feature, we hope to take advantage of affordance learning to score the functional part and provide the optimal contact position. By encoding LTL instructions using Transformer, we want to exploit its representation to understand the task semantics and facilitate the training of the agent. Compared with previous work only using 3D point cloud to guide the agent, when using the affordance module to predict the optimal operating point for the agent, not only can it be used to effectively guide the agent to complete dexterous operations, but also the feedback from its interaction with the environment can update the task representation encoded by Transformer to facilitate the training of the agent, leading to a mutual enhancement.

The goal of the temporal logic guided affordance learning framework is to find an appropriate policy π_φ over the affordance leaning and LTL instructions, such that the desired contact position and LTL representations can guide the agent to achieve effective generalization on dexterous manipulations. The problem of this work can be formally described as follows.

Problem 1. Given a labeled POMDP $\mathcal{M}_e = (S, T, A, p_e, \Pi, L, R, \gamma, \mu, \Omega, O)$ corresponding to task φ, this work is aimed at finding an optimal policy π^*_φ over LTL representation φ_θ, so that the successful rate evaluated on seen and unseen sets of articulated objects under the policy π^*_φ can be maximized.

IV. Method Design

In this section, we propose a novel framework, namely temporal logic guided affordance learning for generalizable dexterous manipulation (TALD), that not only design a contact predictor to guide the agent to manipulate with the max-affordance point observation, but also exploits the task semantics to improve learning efficiency and generalize through category-level evaluation. Section IV-A presents how the vision feature extracted from 3D point clouds can be incorporated into reinforcement learning. Section IV-B explains in detail how to exploit affordance learning to guide the agent generalizing across diverse objects. Section IV-C shows how the LTL representation helps the agent understand the task semantics and improve the agent generalization performance. The overall method behind TALD is illustrated in Fig. 2, which first extracts the vision feature from 3D point clouds, then predicts the max-affordance point observation by the contact predictor and interacts the environment with the guidance of the max-affordance point observation and LTL representation.

A. Feature Extraction and Policy Learning

One of the major challenges in solving problem 1 is the form of visual representation. When RL algorithms are guided by visual information, the observation is usually composed of the images from the camera or the corresponding 3D point cloud. With the continuous improvement of point cloud processing research in the vision field, 3D point cloud RL algorithms not only have better sampling efficiency in the single task [19], but also have better reasoning ability in manipulation generalization [18]–[21]. Therefore, this work mainly focuses on solving the problem of generalization on dexterous manipulation from the view of 3D point cloud.

When 3D point cloud is used to guide the RL algorithm to achieve manipulation generalization, its performance depends on the quality of the observed features extracted by the vision module [18]. Since the point cloud segmentation method based on the functional part of the object [22] can help the agent to reason about the operable region of the unseen object, and provide the necessary foundation for the max-affordance point observation (MPO) [8], we constructed the dataset for the pre-training segmentation following the method provided by [18], which labeled the point cloud into four groups: the functional part of the object, the rest of the object, the robot hand, and the robot arm.

In order to attain generalization at the category level over a range of objects, we use the pre-trained PointNet [23] as an extractor to represent the vision features by inputting 3D point clouds. Specifically, given the observation

o, the PointNet $\mathrm{PN}(\cdot)$ encode the observation o to a vision representation o_{pn}. Then the policy π_e takes the representation o_{pn} as an input and outputs the corresponding action a to interact with the environment. Let $Q_\varphi\left(o_{pn}, a\right)$ and $Q_{\varphi'}\left(o_{pn}^{'}, a^{'}\right)$ be the Q-value function of task φ and $\varphi^{'}$, respectively. Thus in the conventional RL algorithm, such as DQN, the update for Q_φ with the extracted feature can be written as

$$Q_\varphi \leftarrow Q_\varphi + \alpha\left(R_\varphi + \gamma \max_{a'} Q_{\varphi'}\left(o_{pn}^{'}, a^{'}\right) - Q_\varphi\right). \quad (2)$$

By extracting visual features to guide the RL to manipulate the object, the agent has a basic reasoning ability reflected in the segmented functional part of the object. But its reasoning ability can be further improved by the following affordance module.

B. Affordance Learning Optimized by Contact

One of the keys to achieving manipulation generalization is to allow the robot to understand by itself how to manipulate key parts of the object so that it can complete the task quickly and well, i.e. even for objects that it has never seen before. Therefore, another major challenge in solving Problem 1 is how to further determine the optimal manipulation points on the segmented point cloud information. To address this issue, [6] considers affordance information from an interaction perspective and proposes a learning-based framework to estimate per-pixel affordance map for achieving articulated tasks. However, it needs a lot of interactions to get effective samples to update the actionability scoring module for the manipulation with pushing and pulling primitives. So this work hopes to design a contact predictor $\mathrm{CP}(\cdot)$ which improves the quality of the max-affordance point observation (MPO) generated by itself with a high success rate of contact positions during interaction as the RL controller evolves, thus reducing the burden of pre-collection and achieving a similar performance like [6].

The main idea of affordance learning in this work is that the contact predictor selects the max-affordance point observation (MPO) based on the corresponding affordance score map from the object point cloud, and incorporates the MPO as an additional observation o_{MP} to the policy, thus effectively updating the contact predictor based on the contact position of the end-effector s_{eff} of the robot (e.g., the palm of a dexterous hand) and the corresponding success rate of the manipulation.

Specifically, given the object point cloud observation o^{obj}, the contact predictor first extracts the features of each point using PointNet++ [24] and then feeds them to the fully-connected layer (MLP) to generate a map of affordance scores for each point of the action for o^{obj}. Then based on the score of each point, the max-affordance point observation o_{MP} is obtained by averaging the positions of top K scoring points where K is determined by the range of the maximum score. In order to make the end-effector s_{eff} as close as possible to the MPO o_{MP}, we design the max-affordance point reward (MPR) $r_{\mathrm{MPR}} = 1/\left\|o_{\mathrm{MP}} - s_{\mathrm{eff}}\right\|$, which can be viewed as part of r_{env} in (1). Thus the Q-value function $Q_\varphi\left(o_{pn}, a\right)$ is then conditioned on the MPO as $Q_\varphi\left(o_{pn}, o_{\mathrm{MP}}, a\right)$, and (2) can be augmented as

$$Q_\varphi \leftarrow Q_\varphi + \alpha\left(R_\varphi + \gamma \max_{a'} Q_{\varphi'}\left(o_{pn}^{'}, o_{\mathrm{MP}}^{'}, a^{'}\right) - Q_\varphi\right). \quad (3)$$

In order for the contact predictor to predict the optimal o_{MP}, we keep the the MPO o_{MP} generated by the contact predictor as close as possible to the contact point where the end-effector s_{eff} has a high success rate. Let $\mathrm{DGT^i}(s_{\mathrm{eff}})$ indicate the position of interaction with the object i and the contact predictor $\mathrm{CP}(\cdot)$ is updated with $\mathrm{DGT^i}$ as below:

$$\mathrm{CP}^* = \underset{\mathrm{CP}}{\operatorname{argmin}} \sum_i sr^i \left\| \sum_{o_{\mathrm{MP}} \in o^{obj}} \mathrm{CP}(o_{\mathrm{MP}} \mid o_i^{obj}) - \mathrm{DGT^i}(s_{\mathrm{eff}}) \right\|_2 \quad (4)$$

where sr^i is the manipulation success rate on the object i, o_i^{obj} is the point cloud of i-th object and CP^* is the optimal contact predictor.

Algorithm 1 Temporal Logic Guided Affordance Learning

procedure INPUT:(An LTL intruction φ and the POMDP $\mathcal{M}_e$ corresponding to some φ)
  Output: An approximately optimal policy $\pi_\varphi^*(a_t \mid o_t, \varphi)$ for the TL-POMDP $\mathcal{M}_\varphi$
  Initialization: All neural network weights
  Load the pretrained weights to the PointNet $\mathrm{PN}(\cdot)$
  for iteration 1,2,...,N **do** {Exploration Phase}
    for episode 1,2,...,M **do**
      Initialize timer $t \leftarrow 0$ and episode o_0, and augment the observation o_0 with φ_θ encoded by Transformer
      while episode not terminated **do**
        $\varphi^{'} \leftarrow \mathrm{prog}(L(s), \varphi)$
        if $\varphi^{'} \in \{\mathsf{True}, \mathsf{False}\}$ or $s \in T$ **then**
          Break
        end if
        Extract the vision representation o_{pn} from 3D point cloud by the PointNet $\mathrm{PN}(\cdot)$
        Get the max-affordance point observation o_{MP} from object point cloud by the contact predictor $\mathrm{CP}(\cdot)$
        Determine $\widehat{\mathfrak{R}}_\varphi$ by (5) and gather data from φ following π_φ
      end while
    end for
    for training step 1,2,...,K **do** {Training Phase}
      Update all neural network weights by (4) and (6)
    end for
  end for
end procedure

C. LTL Rrepresetation Encoded by Transformer

This section presents how LTL progression can be used to construct a Markovian reward function and how the LTL representation can be encoded from LTL instructions in order to facilitate the robotic learning. In particular, we exploit the LTL progression from [15] and [25] to focus the agent's attention on the current sub-tasks rather than keeping track of the original formula all the time, which takes the LTL instruction φ and the word $\boldsymbol{\sigma} = \sigma_0\sigma_1...$ from the interaction with the environment as inputs and applies the operator $\mathrm{prog}\,(\cdot)$ at step i, $\forall i = 0, 1, \ldots,$ to output the progressed formula following $\mathrm{prog}\,(\sigma_i, p) = \mathrm{True}$ if $p \in \sigma_i$, where $p \in \Pi$ and $\mathrm{prog}\,(\sigma_i, p) = \mathrm{False}$ otherwise. The detailed definition

and examples can be found in [15].

Thus an augmented POMDP corresponding to an LTL instruction φ, namely the task-driven labeled POMDP (TL–POMDP), is proposed by utilizing the LTL progression.

Definition 1. Given a labeled POMDP $\mathcal{M}_e$ with an LTL instruction φ, the TL–POMDP can be built by augmenting $\mathcal{M}_e$ to $\mathcal{M}_\varphi \triangleq \left(\tilde{\mathfrak{S}}, \tilde{\mathfrak{T}}, \tilde{\mathfrak{p}}, \Pi, L, \tilde{\mathfrak{R}}_\varphi, \gamma, \mu, \Omega, O\right)$, where $\tilde{\mathfrak{S}} = S \times \mathrm{cl}(\varphi)$, $\tilde{\mathfrak{p}}((s^{'}, \varphi^{'})|(s, \varphi), a) = p_e(s'|s, a)$ if $\varphi^{'} = \mathrm{prog}(L(s), \varphi)$ and $\tilde{\mathfrak{p}}_i((s^{'}, \varphi^{'})|(s, \varphi), a) = 0$ otherwise, and $\tilde{\mathfrak{R}}_\varphi$ is the reward function corresponding to the instruction φ to address the issue of non-Markovian by exploiting the LTL progression which is

$$\tilde{\mathfrak{R}}_\varphi(s, \varphi) = \begin{cases} r_{env} + r_\varphi, & \text{if } \mathrm{prog}(L(s), \varphi) = \text{True} \\ r_{env} - r_\varphi, & \text{if } \mathrm{prog}(L(s), \varphi) = \text{False.} \\ r_{env}, & otherwise \end{cases} \quad (5)$$

The term $\mathrm{cl}(\varphi)$ refers to the progression closure of φ, which is the smallest set containing φ that is closed under progression.

Compared to converting LTL specifications to automata or RM, our previous work [26] also shows the representation encoded by Transformer [27], which provides flexibility in encoding LTL instructions and facilitate the agent' s training. Given an input $\mathfrak{X}_\varphi = (\mathfrak{x}_0, \mathfrak{x}_1, ...)$ corresponding to the LTL task φ where $\mathfrak{x}_{t,t=0,1,...,}$ denotes the operator or proposition in sc-LTL, $\mathfrak{X}_\varphi$ will be embedded by the word embedding E following $\mathfrak{X}_\mathrm{E} = [\mathfrak{x}_0\mathrm{E}; \mathfrak{x}_1\mathrm{E}; ...; \mathfrak{x}_N\mathrm{E}] \in \mathbb{R}^{B \times M \times D}$ where B is the batch size, M is the length of input $\mathfrak{X}_\varphi$ and D is the model dimension of Transformer, and then $\mathfrak{X}_\mathrm{E}$ is combined with the frequency-based positional embedding E_{pos} to take use of the sequence's ordering. The outline of LTL Encoder for TALD is shown in Fig. 2. By representing the LTL instruction in TL-POMDP, we can effectively utilize it to improve the agent's learning efficiency and guide the agent with the task semantics.

By representing the LTL instruction φ via φ_θ encoded by Transformer, $Q_\varphi\left(o_{pn}, o_{\mathrm{MP}}, a\right)$ can be revised as $Q_{\varphi_\theta}\left(o_{pn}, o_{\mathrm{MP}}, a\right)$ and (3) can be augmented as

$$Q_{\varphi_\theta} \leftarrow Q_{\varphi_\theta} + \alpha\left(\tilde{R}_\varphi + \gamma \max_{a^{'}} Q_{\varphi^{'}_\theta}\left(o^{'}_{pn}, o^{'}_{\mathrm{MP}}, a^{'}\right) - Q_{\varphi_\theta}\right). \quad (6)$$

In this way, the agent can incorporate the task semantics by encoding the corresponding representation φ_θ and try to manipulate the functional parts of objects guided by the max-affordance point observation.

The pseudo-code is outlined in Alg. 1. Before the training, TALD first load the pretrained weights to PointNet $\mathrm{PN}(\cdot)$ (line 2). In the exploration stage, TALD applies the operator $\mathrm{prog}\left(\cdot\right)$ to progress the LTL formula φ and encode the LTL representation φ_θ(lines 5-10). Then TALD exploits the $\mathrm{PN}(\cdot)$ to extract the vision feature $o_{pn}$from 3D point cloud (line 11) and predict the max-affordance point observation $o_{MP}$by averaging the positions of top K scoring points (line 12). By input above information into the policy π_φ, TALD interacts with the action a from π_φ to get the reward $\tilde{\mathfrak{R}}_\varphi$ (line 13). In the update stage, TALD updates all neural network weights by (4) and (6) (lines 16-17).

V. CASE STUDIES

In this section, we compare the proposed TALD framework against baselines in terms of learning efficiency and generalization performance over four scenarios. In particular, we consider the three aspects as follows. **1) Performance:** How well does our approach outperform baselines in four dexterous manipulation scenarios? **2) Affordance:** What is the role of affordance learning for TALD? **3) Representation:** How well can the LTL representation help the agent improve the generalization via Transformer?

A. Baselines and Tasks Setting

Baselines. In order to demonstrate the effectiveness of TALD, it is empirically compared against four baselines. The first baseline is DexArt from [18] which proposes a general framework for dexterous manipulation over articulated objects and generalizes well on different kinds of objects by exploiting RL. The second baseline is $\mathrm{DexArt}_{\mathrm{afford}}$ which augments the DexArt by guiding the agent with affordance learning. The third baseline is our TALD without the considering max-affordance point observation MPO, denoted by $\mathrm{TALD}^{\mathrm{MPO}}_{\mathrm{w/o}}$, which aims to explore the help of MPO for TALD. The fourth baseline is our TALD without considering the max-affordance point reward (MPR), denoted by $\mathrm{TALD}^{\mathrm{MPR}}_{\mathrm{w/o}}$, which aims to study the importance of MPR for the policy learning of TALD.

Tasks Setting. To effectively evaluate the performance of above algorithms, four challenging generalizable dexterous manipulations in [18] are selected and the corresponding LTL instructions are listed in our project. To guarantee the fairness of the comparison, we choose PPO [28] as the backbone of RL like [18] over 3 seeds.

B. Main Simulation Results

1) Success Rate. Table. I shows the success rate (mean $\pm$ std) of different methods on four scenarios for both seen and unseen objects. It can be observed that (1) algorithms with the help of affordance module ($\mathrm{DexArt}_{\mathrm{afford}}$ and TALD) can get a higher success rate than DexArt, especially when generalizing to the unseen objects in the Faucet and Toilet tasks, having 8%~25% improvement; (2) Based on LTL representation encoded by Transformer, TALD can outperform other methods and get more stable generalization performance in most of the scenarios, since the task module can extract the task semantics from the LTL instructions and facilitate the robotic learning with the LTL representation described in Sec. IV-C; and (3) TALD only perform poorly than DexArt on the seen objects of the Laptop task and we speculate that DexArt is overfitting on the laptop's seen set, leading to its mediocre performance on the unseen set.

2) Ablation Study. To show the importance of designs of MPO and MPR on TALD, we evaluate the ablation studies

Fig. 3: We illustrate the segmentation based on the 3D point clouds and the max-affordance point observation obtained by averaging the positions of top K scoring points in the Toilet and Laptop tasks.

TABLE I: Success Rate of Different Methods on four scenarios for Both Seen and Unseen Objects.

Task	Faucet		Bucket		Laptop		Toilet	
Split	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
DexArt	0.790 ± 0.020	0.580 ± 0.070	0.750 ± 0.040	0.760 ± 0.070	**0.920 ± 0.020**	0.600 ± 0.070	0.850 ± 0.010	0.550 ± 0.010
$\text{DexArt}_{\text{afford}}$	0.760 ± 0.060	0.624 ± 0.182	0.509 ± 0.202	0.563 ± 0.207	0.542 ± 0.048	0.533 ± 0.000	0.898 ± 0.051	0.597 ± 0.063
TALD	0.803 ± 0.042	**0.728 ± 0.023**	**0.751 ± 0.030**	**0.792 ± 0.033**	0.660 ± 0.135	**0.650 ± 0.098**	**0.933 ± 0.010**	**0.639 ± 0.026**
$\text{TALD}^{\text{MPO}}_{\text{w/o}}$	**0.841 ± 0.014**	0.664 ± 0.007	0.388 ± 0.282	0.342 ± 0.256	0.403 ± 0.144	0.378 ± 0.150	0.931 ± 0.010	0.627 ± 0.030
$\text{TALD}^{\text{MPR}}_{\text{w/o}}$	0.805 ± 0.108	0.681 ± 0.166	0.176 ± 0.248	0.146 ± 0.207	0.630 ± 0.227	0.556 ± 0.228	0.926 ± 0.007	0.612 ± 0.023

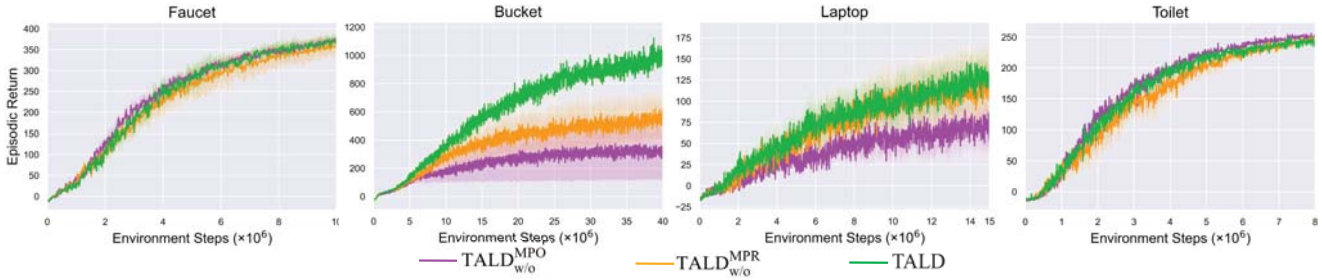

Fig. 4: The reward performance shows the effect about the design of the max-affordance point observation and the max-affordance point reward during the training stage.

$\text{TALD}^{\text{MPO}}_{\text{w/o}}$ and $\text{TALD}^{\text{MPR}}_{\text{w/o}}$, and show the corresponding results in Fig. 4 and Table. I. As shown in Fig. 4 and Table. I, it can be observed that (1) MPO has a bigger effect on the performance of TALD, which is obvious on the Bucket and the unseen objects of Faucet scenarios; (2) even without the help of MPO, $\text{TALD}^{\text{MPO}}_{\text{w/o}}$ can show a higher success rate on the seen objects of Faucet tasks, which is about 45% higher than Dexart; (3) the design of MPR, on the other hand, does not contribute much to the generalization performance of TALD, but it is important for the Buckets scenario; (4) by evaluating the success rates of $\text{TALD}^{\text{MPO}}_{\text{w/o}}$ and $\text{TALD}^{\text{MPR}}_{\text{w/o}}$, showing the design of the max-affordance point observation and the max-affordance point reward greatly improves the generalization effect of the Bucket scenario; as well as (5) $\text{TALD}^{\text{MPO}}_{\text{w/o}}$ and $\text{TALD}^{\text{MPR}}_{\text{w/o}}$ outperform DexArt on all sets of Faucet and Toilet tasks, which further ensure the contribution of affordance learning and task module to the performance of the framework.

3) MPO Visualization. To show the help of the affordance module in TALD, we illustrate the segmentation based on the 3D point clouds and the max-affordance point observation obtained by averaging the positions of top K scoring points in Fig. 3, whose red part represents the segmentation corresponding to the functional part and green points means the MPO derived from the affordance map. Note that the angle of the simulation view in Fig.3 is not necessarily the angle at which the camera captured the point cloud. We show it this way only to clearly show the corresponding simulation scene.

C. Limitations

Simulation results present that, with the design of contact predictor and LTL Encoder in TL-POMDP, TALD can improve learning efficiency during the training and generalization performance when evaluating on unseen objects. While TALD has great potential for generalization better on dexterous manipulations, there remain several issues. The first issue is that the RL backbone of TALD is PPO, whose distribution is usually unimodal. This means that it may lack expressiveness in complex environments and generalize poorly in challenging scenarios such as the laptop task. More effective RL algorithms with a multimodal distribution policy are helpful in solving the above problem. Another potential issue is that the LTL Encoder used for

the task module is inspired from the original Transformer [27]. With the development of the family of Transformer, more effective designs of architecture will further improve the method's performance.

VI. CONCLUSIONS

In this work, we propose a novel method, TALD, to enable efficient manipulation of articulated objects and greatly improve the category-level generalization performance of dexterous manipulations. In particular, LTL Progression is used to construct TL-POMDP to avoid non-Markovian reward functions. And TALD is equipped with affordance learning for predicting actionable information and LTL representation for understanding the task semantics and improving the learning efficiency. We validate the proposed method in four articulated manipulation scenarios. The generalization performance corresponding to success rates and visualization effects show the effectiveness of TALD. Future work will consider optimizing robotic performance to behave more like humans and deploying it in real-world experiments.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[3] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, "First return, then explore," *Nature*, vol. 590, no. 7847, pp. 580–586, 2021.

[4] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[5] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *Proc. IEEE Int. Conf. Rob. Autom.* IEEE, 2022, pp. 6372–6378.

[6] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proc. IEEE Int. Conf. Comput. Vision.*, 2021, pp. 6813–6823.

[7] Y. Zhao, R. Wu, Z. Chen, Y. Zhang, Q. Fan, K. Mo, and H. Dong, "Dualafford: Learning collaborative visual affordance for dual-gripper manipulation," in *Proc. Int. Conf. Learn. Representations*, 2022.

[8] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, "Rlafford: End-to-end affordance learning for robotic manipulation," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2023, pp. 5880–5886.

[9] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, 2007.

[10] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[11] Z. Zhou, Z. Chen, M. Cai, Z. Li, Z. Kan, and C.-Y. Su, "Vision-based reactive temporal logic motion planning for quadruped robots in unstructured dynamic environments," *IEEE Trans. Ind. Electron.*, 2023.

[12] H. Wang, H. Zhang, L. Li, Z. Kan, and Y. Song, "Task-driven reinforcement learning with action primitives for long-horizon manipulation skills," *IEEE Trans. Cybern.*, 2023.

[13] X. Li, Z. Serlin, G. Yang, and C. Belta, "A formal methods approach to interpretable reinforcement learning for robotic planning," *Sci. Robot.*, vol. 4, no. 37, 2019.

[14] M. Cai, M. Hasanbeig, S. Xiao, A. Abate, and Z. Kan, "Modular deep reinforcement learning for continuous motion planning with temporal logic," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7973–7980, 2021.

[15] H. Zhang and Z. Kan, "Temporal logic guided meta q-learning of multiple tasks," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8194–8201, 2022.

[16] P. Vaezipoor, A. Li, R. T. Icarte, and S. McIlraith, "Ltl2action: Generalizing ltl instructions for multi-task rl," *arXiv preprint arXiv:2102.06858*, 2021.

[17] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Form. Methods Syst. Des.*, vol. 19, no. 3, pp. 291–314, 2001.

[18] C. Bao, H. Xu, Y. Qin, and X. Wang, "Dexart: Benchmarking generalizable dexterous manipulation with articulated objects," in *Proc. - IEEE Conf. Comput. Vis. Pattern Recognit, CVPR*, 2023, pp. 21 190–21 200.

[19] Z. Ling, Y. Yao, X. Li, and H. Su, "On the efficacy of 3d point cloud reinforcement learning," *arXiv preprint arXiv:2306.06799*, 2023.

[20] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," *arXiv preprint arXiv:2107.14483*, 2021.

[21] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[22] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang, "Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," in *Proc. - IEEE Conf. Comput. Vis. Pattern Recognit, CVPR*, 2023, pp. 7081–7091.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. - IEEE Conf. Comput. Vis. Pattern Recognit, CVPR*, 2017, pp. 652–660.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. neural inf. process. syst*, vol. 30, 2017.

[25] F. Bacchus and F. Kabanza, "Using temporal logics to express search control knowledge for planning," *Artif Intell*, vol. 116, no. 1-2, pp. 123–191, 2000.

[26] H. Zhang, H. Wang, and Z. Kan, "Exploiting transformer in sparse reward reinforcement learning for interpretable temporal logic motion planning," *IEEE Robot. Autom. Lett.*, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. neural inf. process. syst*, vol. 30, 2017.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.