Learning Multi-Task Policies for Robotics

Elliot Chane-Sane

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Learning Multi-Task Policies for Robotics

Soutenue par
Elliot Chane-Sane
Le 6 Septembre 2023

École doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité
Informatique

Composition du jury :

Fabien MOUTARDE Mines Paris	*Président du jury* *Examinateur*
Christian WOLF Naver Labs Europe	*Examinateur*
David FILLIAT ENSTA Paris	*Rapporteur*
Nicolas MANSARD LAAS	*Rapporteur*
Ivan LATPEV Inria	*Directeur de thèse*
Cordelia SCHMID Inria	*Directeur de thèse*

ENS | PSL

Résumé

Le développement de robots généralistes capables d'accomplir une vaste gamme de tâches présente un énorme potentiel pour alléger la charge de travail humain dans des tâches physiquement exigeantes, dangereuses ou fastidieuses. Malgré les études récentes sur l'utilisation de l'apprentissage profond pour le contrôle des robots, dans l'espoir de réaliser des avancées comparables à celles observées dans des domaines tels que la vision ou la compréhension du langage, le développement de robots polyvalents de cette nature est demeuré une entreprise complexe à concrétiser. Le contraste entre les progrès de l'apprentissage robotique par rapport à d'autres domaines de l'intelligence artificielle peut être attribué au manque de données disponibles sur Internet pouvant être directement exploitées pour l'apprentissage des politiques. De surcroît, le processus de collecte et d'annotation de données spécifiquement destinées à la robotique est chronophage et spécifique à une configuration matérielle. Dans cette thèse, nous introduisons de nouvelles méthodes pour l'apprentissage de politiques robotiques multitâches, offrant des solutions prometteuses pour relever ces défis.

Notre première contribution est un nouvel algorithme d'apprentissage par renforcement qui apprend des politiques de contrôle robotique en interagissant avec l'environnement pour atteindre des configurations souhaitées. Le fait de formuler une tâche comme une configuration à atteindre présente l'avantage de permettre la définition de nombreuses tâches sans nécessiter de définir des fonctions de récompense spécifiques pour chaque tâche. Notre approche consiste à entraîner une politique de haut niveau qui utilise des sous-objectifs imaginés pour guider l'apprentissage, lesquels sont ensuite abandonnés après l'achèvement de l'entraînement. Notre approche démontre une meilleure efficacité d'échantillonnage et peut résoudre des tâches temporellement étendues plus complexes, surpassant les travaux antérieurs sur des problèmes de locomotion complexes et de manipulations à partir d'observations visuelles.

Définir une tâche au travers de configurations à atteindre limite l'éventail de tâches possibles et peut poser des difficultés pour les utilisateurs non experts. Notre deuxième contribution est une méthode d'apprentissage de politiques capables de suivre des instructions vidéo humaines pour la manipulation robotique multitâche basée sur la vision. Cette approche offre une manière plus accessible de communiquer la tâche souhaitée au

robot, tout en ouvrant la possibilité de généralisation à des tâches pour lesquelles la politique n'a pas été spécifiquement entraînée. À partir de trajectoires de robot collectées au préalable, nous entraînons une politique multitâche en la conditionnant sur les vidéos correspondants aux séquences d'images capturées par la caméra du robot. Ensuite, nous conditionnons la politique avec une démonstration vidéo humaine en utilisant une fonction de similarité entre les vidéos apprises à partir d'une vaste collection de vidéos humaines. Notre approche permet le contrôle des robots à l'aide de démonstrations humaines de manière non supervisée, car nous n'utilisons pas de trajectoires de robot appariées à des instructions humaines lors de l'entraînement. Cela élimine le processus fastidieux d'annotation humaine ou de conception de récompenses pour chaque tâche robotique que nous souhaitons aborder.

Mots clés : apprentissage profond, apprentissage par renforcement, vision par ordinateur, robotique

Abstract

Developing generalist robots capable of performing a wide range of tasks holds great potential for minimizing human labor in physically demanding, dangerous, and tedious tasks. Although deep learning has been used in robot control in recent years, with the goal of achieving similar advancements as those observed in fields such as vision or language understanding, the development of such versatile robots has remained elusive. The discrepancy between the progress of robot learning compared to other domains of machine learning can be attributed to the scarcity of readily available data from the internet that can be directly utilized for policy learning. Additionally, the process of collecting and annotating data specifically for robots is time-consuming and specific to a hardware configuration. In this thesis, we introduce novel methods for learning multi-task robot policies, offering promising solutions to address these challenges.

Our first contribution is Reinforcement learning with Imagined Subgoals (RIS), a novel reinforcement learning algorithm that learns goal-reaching policies through online interactions with its environment to achieve desired goal configurations. Formulating tasks with the goal to reach has the advantage of allowing for a wide range of tasks to be defined without the need for task-specific rewards. Our approach involves training a high-level policy that leverages imagined subgoals to guide policy learning, which are subsequently discarded after training completion. Our approach demonstrates higher sample efficiency and the ability to solve more complex temporally extended tasks, outperforming prior works on challenging robotic locomotion and vision-based manipulation tasks.

Using only goal configurations to define tasks limits tasks to goal-reaching objectives and may pose challenges for non-expert users to provide specific configurations. Our second contribution is Video-conditioned Policy learning (ViP), a method for learning robot policies that can follow human video instructions for multi-task vision-based robotic manipulation. This approach provides a user-friendly way for non-experts to communicate the desired task to the robot, goes beyond basic goal-reaching skills, and opens up opportunities to generalize to tasks that the policy was not specifically trained on. Using offline robot trajectories, we train a multi-task policy by incorporating corresponding robot videos, capturing the sequence of camera images, as conditioning information. Next, we prompt the policy with a human video demonstration, leveraging a similarity function

between videos learned from a large collection of human videos. Our approach enables robot control by human demonstrations in a zero-shot manner : we don't rely on paired robot trajectories and human instructions during training, eliminating the tedious process of meticulous human annotation or reward shaping for every robot task we aim to tackle.

Keywords : deep learning, reinforcement learning, computer vision, robotics

Contents

Chapter 1

Introduction

The last decade has witnessed significant advances in deep learning, which have transformed the fields of artificial intelligence and machine learning. In particular, deep learning has been instrumental in pushing the state-of-the-art in image classification (Krizhevsky et al., 2012), segmentation (Long et al., 2015) and object detection (Ren et al., 2015). In natural language processing, it has also been used to generate realistic and coherent text, opening up new possibilities in natural language generation and language modeling (Vaswani et al., 2017; Brown et al., 2020). Other important applications of deep learning include speech processing (Hannun et al., 2014), recommender systems (Covington et al., 2016), protein discovery (Jumper et al., 2021) and many more.

This progress has been driven by the ability to train deep neural networks (LeCun et al., 2015), a machine learning model composed of multiple layers of interconnected artificial nodes employed as function approximators, on large and diverse datasets. This approach has been made possible by the availability of large datasets, notably from the internet (Deng et al., 2009; Grauman et al., 2022; Raffel et al., 2020), and the increased compute power available to train more complex and deeper neural networks (He et al., 2016). By doing so, the networks can discover patterns within the data that arise from its abundance and diversity and that are applicable to unseen inputs and tasks, which is known as generalization. These developments have provided exciting avenues for research and innovation, propelling the field of artificial intelligence towards new breakthroughs that have already deeply impacted people's lives in numerous ways.

Motivated by this success, recent research in robotics aims to leverage the progress in machine learning for robot control. Using deep neural networks for control policies and decision making holds the promise of creating generalist robotic agents with intelligence and flexibility to adapt to changing situations and perform tasks that were not explicitly programmed. Indeed, a wide variety of robots that are capable of performing complex skills, including arm robots such as Franka Emika, quadruped robots like Boston Dy-

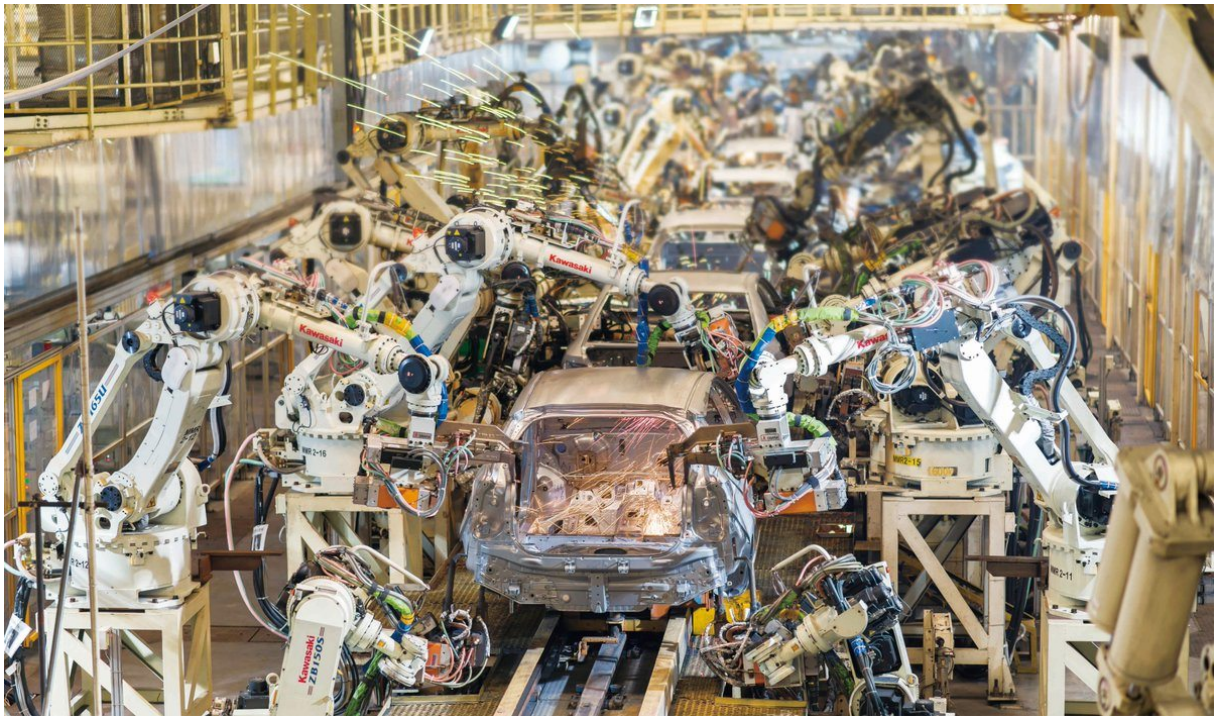


Figure 1.1 – Toyota started automating its car factories with arm robots in the 1970s. To this day, robots predominantly operate in controlled environments, carrying out relatively straightforward and repetitive tasks.

namics's Spot, mobile manipulators such as PAL Robotics' TIAGo and humanoid robots such as Honda's Asimo, have been available for a long time (Figure 1.2). However, traditional robotic control has restricted the use of robots to simple and repetitive tasks in controlled environments, such as those found in factories (Figure 1.1), agriculture and logistics. Robot learning has the potential to build more sophisticated robots capable of performing a wider range of tasks in cluttered and uncontrolled environments, sometimes operating alongside people. This has the potential to greatly benefit society by reducing the workload of humans for tasks that are repetitive, tedious or dangerous.

As of today, however, robot learning has not yet experienced the same level of success as other areas of machine learning. Certain advancements in deep learning, notably in object and scene representations (Mildenhall et al., 2021; Labbé et al., 2020; Strudel et al., 2021) can contribute to the accuracy of the existing robotics pipelines (Siam et al., 2017; Hong et al., 2018; Manuelli et al., 2019; Ichnowski et al., 2021). However, these have not directly resulted in the creation of generalist robots. Perhaps the most challenging aspects of robot learning is decision-making for robots. In robot learning, robot policies are trained to predict the optimal commands to be sent to the actuators at each time step, aiming for desired future outcomes. The goal is to generate an appropriate sequence of actions that effectively leads to the desired changes in the world over time. To this day,

Figure 1.2 – Examples of available robot platforms that can physically perform various skills. From left to right: Franka Emika,Boston Dynamics's Spot, PAL Robotics' TIAGo and Honda's Asimo

learning generalizable control policies for robotics remains an open problem fraught with various challenges. As a result, current robot learning systems have mainly been confined to controlled laboratory environments, lagging behind other fields which have seen greater advancements and real-world applications such as computer vision and natural language processing.

1.1 Goal

The goal of this thesis is to propose novel algorithms for learning control policies for robots: systems that process robot observations and generate suitable actions for the robot to execute in its environment. Robot observations will typically correspond to camera inputs and proprioceptive inputs, i.e. internal states or sensor measurements of the robot itself such as joint angles, velocities or torques inputs, but could also include other types such as tacticle, lidar or audio observations.

Our focus will center on three aspects:

1. multi-task robot learning,
2. long-horizon reasoning,
3. and following human instructions.

As we will see, these three points are intertwined. Indeed, while some tasks are simple for

robots to execute, many other tasks require long-horizon planning. Multi-task learning facilitates the process of temporally extended reasoning by enabling the robot to chain together multiple tasks to achieve more complex tasks. Moreover, following human instructions serves as a valuable means to define tasks and provides a practical framework for skill abstraction.

1.1.1 Multi-task robot learning

Our research focuses on multi-task policy learning, where the policy takes into account not only observations but also task instructions to perform the desired skills. We are working towards building generalist robot controllers that can adapt and generalize to new tasks, expanding their capabilities beyond specific skills.

Traditional single-task learning methods are limited in their applicability to handle multiple tasks, as they are designed to optimize performance for a specific task. Multi-task learning seeks to address this limitation by jointly learning from multiple tasks, enabling robots to transfer knowledge and skills across related tasks. By leveraging the shared structure and dependencies among tasks, multi-task learning can enhance the overall performance of the robot across various domains. While single task robot learning has been extensively studied, multi-task learning introduces distinct challenges. These challenges include (i) designing a single policy that can adapt to different tasks and (ii) effectively communicating task instructions to the robot. However, multi-task learning also opens up exciting possibilities for advancing robot learning methods and enhancing their capabilities in handling diverse tasks.

In multi-task robot learning, the representation and integration of task instructions play a crucial role. One approach is to collect data for each task we want to solve and represent them using one-hot vectors, but this approach limits the generalization to new tasks and requires expert data collection. Another option is to use natural language instructions, but it often requires labeled demonstrations to train the policy or reward function. In this thesis, one of our focuses is on learning goal-reaching policies, where the task is defined as reaching a desired goal within the environment. At inference time, the human user expresses their expectations by providing a desired configuration, allowing the policy to encode multiple tasks and generalize to new ones. By leveraging autonomous data generation and user-defined goals, our approach enables the policy to handle a wide range of tasks, eliminating the need for task-specific demonstrations and enhancing its adaptability and versatility.

1.1.2 Long-horizon reasoning

Long-horizon reasoning is a key challenge in robotics, as it requires the robot to plan and execute complex tasks that span multiple time steps involving many subtasks. This is crucial for tasks that require a series of actions to achieve a long-term goal, such as cooking a meal or assembling a complex object. The concept of temporally extended reasoning in robot learning is often broken down into low-level and high-level reasoning. Low-level reasoning involves basic motor skills such as walking or grasping, while high-level reasoning involves planning and decision-making to achieve long-term goals, such as navigating through a cluttered environment or assembling a complex object.

Consider the task of making a salad, which involves a sequence of small steps such as grasping utensils, tomatoes, and lettuce leaves, moving objects, and pouring sauce onto the plate. Each of these small steps corresponds to low-level skills. Deciding the appropriate order of these skills represents high-level reasoning. Furthermore, even within the scope of low-level skills, each skill can be decomposed further. For example, grasping an object entails moving the robot arm towards a specific grasping zone, closing the gripper to secure the object, and then lifting it. This decomposition allows for finer control and coordination of the robot's actions. Additionally, completing a single plate of salad can be viewed as a substep towards achieving a larger goal, such as serving a full-course dinner to an entire family.

We aim to propose learning algorithms that empower robots to efficiently handle more complex tasks by performing such hierarchical decompositions. We explore novel methods for skill abstraction to decompose complex tasks into simpler ones without the need for additional human guidance or supervision, enabling robots to autonomously acquire and refine their task decomposition abilities.

In this context, multi-task policy learning can be instrumental in addressing long-horizon settings by providing skill abstractions. By encapsulating a variety of tasks within a single policy, it offers a low-level policy that can be effectively prompted by a high-level planning method to solve complex, temporally extended tasks. In this thesis, we propose a novel method that leverage the compositionality of goal-reaching tasks to train long-horizon polices.

1.1.3 Following human video instructions

Another objective of our thesis is to define tasks as a human video instruction to follow. Indeed, compared to specifying a desired goal in the environment, conveying the intended task through a human video demonstration requires less expertise and offers a more user-friendly solution for non-expert users. Moreover, this could open up new opportunities for

temporally extended planning by chaining human instructions. In contrast to previous studies that focused on imitating a single human video, our research tackles a more practical scenario where the task to be performed is unknown to the policy during training and video prompts can be collected from diverse and natural settings, reflecting real-world conditions.

We are motivated by the potential of harnessing extensive video datasets in robotics to enhance the learning of new tasks. A key challenge arises when a robot needs to accomplish a task for which it lacks demonstrations or explicit training. In such cases, video data showcasing manipulation skills becomes a promising resource for conveying how the task can be performed. We leverage a video encoder pretrained on human videos and demonstrate its effectiveness in two important aspects: encoding diverse robot behaviors and mapping video prompts to corresponding robot behaviors. Our approach enhances the robot's ability to perform tasks for which it has not been explicitly trained.

1.2 Challenges

Deep learning for decision making in robotics encounters distinct challenges that stem from the physical constraints inherent to robots. Unlike purely virtual systems, robots must operate in the real world, where they are subject to limitations such as limited sensing capabilities, motor constraints, and physical interactions with the environment. These constraints introduce complexities that require careful consideration in the design and application of deep learning algorithms. This section delves deeper into the specific challenges we face in our study.

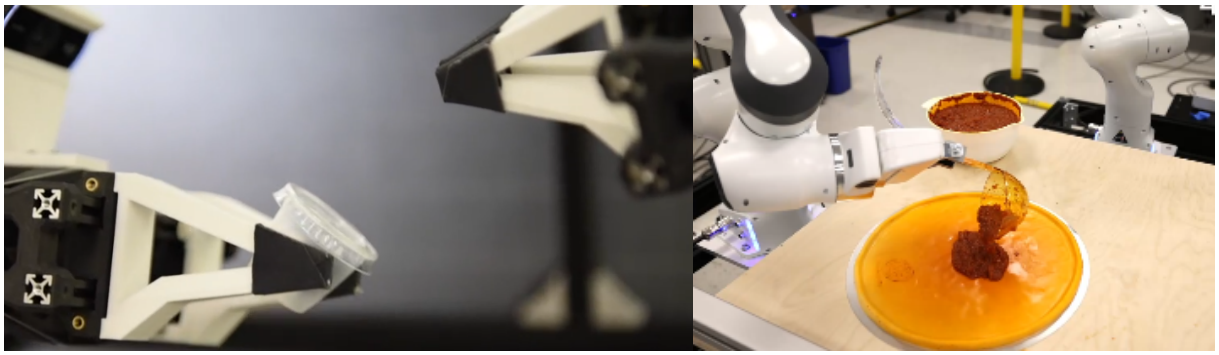


Figure 1.3 – Everyday manipulations tasks, such as opening a lid (left) or pouring sauce (right), while simple for humans, are still daunting for robots as they require precise manipulation and feedback control from cameras.

Figure 1.4 – Training reinforcement learning algorithms on real robots typically requires a large number of interactions, careful design of reward functions, and automatic reset mechanisms. Despite these efforts, the resulting policies are often confined to controlled lab environments.

1.2.1 Lack of data for imitation learning

Perhaps the simplest way to learn robot policies is through supervised imitation learning, where the robot learns to mimic the behavior of demonstrations by directly mapping observations to actions. While this approach can learn complex robot skills (Figure 1.3), the lack of access to large and diverse high-quality datasets for imitation learning has impaired progress in robot learning. Unlike for instance in computer vision or natural language processing, where billions of image and text documents as well as decades of videos are uploaded on the internet daily, the process of collecting data for robot learning is much more tedious. Indeed, collecting robot demonstrations often requires human intervention to program and record the data. This can involve setting up different scenarios, environments, and situations for the robot to interact with and collecting data from various sensors and cameras. Data collection for imitation learning can be approached in two ways: through human teleoperation, where a human operator directly controls the robot to gather demonstrations, or by designing scripts to automatically collect data for each task. However, both methods necessitate substantial human effort for data collection and labeling. As a result, for instance, the largest collections of demonstrations available for robot control (Bennetts et al., 2016) currently pales in comparison to the scale and diversity of typical datasets used to benchmark progress in computer vision tasks (Deng et al., 2009; Lin et al., 2014; Kay et al., 2017).

Figure 1.5 – Recent visually realistic simulators, such as OmniGibson (Li et al., 2023), offer intermediate solutions to train policies on complex and diverse environments with many objects.

1.2.2 Online data collection with reinforcement learning

Automatically collecting data with online reinforcement learning provides an alternative approach where the robot can gather data by exploring and interacting with its environment, and the policy can be learned by maximizing cumulative rewards obtained through its actions. However, reinforcement learning comes with its own set of challenges (Figure 1.4). Reinforcement learning often requires tremendous amount of interactions to learn good policies. Yet robots have limitations on the amount of data they can collect. Additionally, the design of effective reward functions for learning optimal policies in reinforcement learning often relies on extensive human expertise. Sparse reward functions, which provide positive rewards only upon task completion, present difficulties for reinforcement learning in terms of exploration and the ability to reason over long horizons.

1.2.3 Limits of simulations

Robot simulators offer an appealing intermediate solution for data collection in robotics due to their ability to facilitate large-scale data generation for both imitation and reinforcement learning (Figure 1.5). In addition, they bypass reset and safety issues and provide efficient way to benchmark progresses in robot learning. They also provide access to detailed world state information during training. However, creating realistic simulators for robotics poses significant challenges due to the complexity of accurately modeling the physics and visual appearances of the world (Erez et al., 2015; Lidec et al., 2023). Due to

Figure 1.6 – Using large and diverse datasets featuring human video demonstrations of manipulation skills, like the Something-Something dataset (Goyal et al., 2017), holds potential for robot learning. However, connecting human videos to robots presents significant challenges.

the presence of this reality gap (Jakobi et al., 1995), policies trained exclusively in simulation often demonstrate subpar performance when deployed on real-world robotic systems (Tan et al., 2018). Moreover, constructing rich simulated environments that encompass a wide range of objects and accurately represent everyday scenarios requires substantial effort and resources. Despite the benefits of using simulators, data collection for imitation learning or with reinforcement learning still necessitates human efforts.

1.2.4 Robot learning with human data

Another approach in robot learning involves incorporating human data such as human videos (Figure 1.6), considering that human children learn locomotion and manipulation skills by observing others. However, this approach faces two challenges. Firstly, there is often an embodiment gap between humans and robots, as robots may have different physical characteristics and capabilities compared to humans, such as the use of wheels or a different number of arms and fingers. Even robots designed to resemble humans are inherently constrained by the limitations of their physical bodies, which differ from the capabilities of humans. These constraints impose restrictions on the range of actions and movements that humanoid robots can perform, making their possibilities more limited

compared to humans. Furthermore, when it comes to acquiring human data that provides precise and explicit instructions for manipulations or locomotions, such as grasping objects or opening bottles, the availability of such information on the internet is quite limited. While there is an abundance of videos showcasing various activities, such as cooking and following recipes, the focus of these resources tends to be on higher-level skill abstractions, assuming that low-level human control is already understood. Unfortunately, the specific details and nuances required for performing fine-grained manipulation tasks are often lacking. This gap in readily available data hinders the ability to effectively learn control policies for robotics, as the crucial information needed for training these policies remains elusive.

1.3 Contributions

This thesis is based on the following two publications:

— Elliot Chane-Sane, Cordelia Schmid and Ivan Laptev. Goal-Conditioned Reinforcement Learning with Imagined Subgoals. In ICML 2021. (Chane-Sane et al., 2021)

— Elliot Chane-Sane, Cordelia Schmid and Ivan Laptev. Learning Video-Conditioned Policies for Unseen Manipulation Tasks. In ICRA 2023. (Chane-Sane et al., 2023)

1.3.1 Long-horizon reasoning for goal-conditioned reinforcement learning

In Chapter 3, we propose a new goal-conditioned reinforcement learning algorithm to solve tasks that require temporally extended reasoning in a more sample efficient manner. Given the current state and a desired goal configuration, the policy must predict the actions that could lead the agent to reach the goal in the future. This approach is particularly valuable in multi-task learning for robotics, as it eliminates the need for additional human expertise to define task-specific rewards. However, reinforcement learning often struggles to solve tasks that require temporally extended reasoning.

Our approach leverages the compositionality of goal-reaching tasks to guide policy learning towards long-horizon goals with simpler imagined subgoals during training. Our imagined subgoals are intermediate states halfway to the goal predicted by a separate high-level policy. This high-level policy is trained jointly with the policy. To predict appropriate imagined subgoals, it uses the value function of the policy as a reachability metric. During training, the policy is encouraged to reach these subgoals implicitly: we use imagined subgoals to define a prior policy, and incorporate this prior into a KL-constrained policy iteration scheme to speed up and regularize goal-conditioned reinforcement learning. The

high-level policy and its imagined subgoals are discarded at test time, where the learned policy can achieve long-horizon goals on its own, eliminating the need for hierarchical decomposition at test time. We evaluate our approach on complex robotic locomotion and manipulation tasks and show that it outperforms prior methods by a large margin.

1.3.2 Learning policies to follow human videos

Despite the advantages it brings, there are inherent limitations to using a desired goal as the sole means of conveying the intended task to the robot. In Chapter 4, we present a method to learn robot policies to follow human video instructions in multi-task settings. Given a video of a human performing a manipulation task in a natural environment, the robot must execute a similar skill within a multi-task environment. This approach offers a user-friendly means for non-experts to convey the desired task to the robot, extending beyond simple goal-reaching skills and opening up opportunities to generalize to tasks that the policy was not specifically trained on.

To this end, we propose to learn a policy that uses its own robot videos as supervision during training: given a dataset of robot trajectories, the policy predicts the action given the observation and a video of the full robot trajectory. More precisely, we use a video encoder pretrained for human action recognition on a large and diverse dataset of human videos. Video embeddings of the robot trajectories establish an embedding space that functions as a means of skill abstraction, allowing the policy to be trained in performing a variety of behaviors. At test time we retarget human videos to this robot embedding space to perform a skill similar to the human video prompt. Our approach enables robot control by human demonstrations in a zero-shot manner, i.e. without using robot trajectories paired with human instructions during training, alleviating the need for careful human annotation or reward design for each robot task we intend to solve.

1.4 Outline

This thesis consists of 5 chapters including this introduction. In Chapter 2, we review previous work in robot learning most related to the work presented in this thesis. We focus our review on learning control policies using imitation and reinforcement learning, on perception-based methods for robotic manipulation and on robot learning with human videos. In Chapter 3, we present our method to learning goal-conditioned policies with reinforcement learning that can solve temporally extended tasks. In Chapter 4, we propose a novel method to map human video prompts to robot skills. We conclude in Chapter 5 with a summary of contributions, limitations and a discussion of promising future work directions.

Chapter 2

Related Work

This chapter is structured into two sections, providing an overview of relevant literature on learning control policies for robotics and exploring the field of robot learning using human data. In Section 2.1, we present approaches to learn deep neural policies for robotic control. These policies can be learned either by imitation or reinforcement learning, with hierarchical reinforcement learning offering a potential solution for addressing the complexities of long-horizon reasoning. In Section 2.2, we present robot learning methods that use human data. We present an overview of seminal works in these domains, highlighting their current limitations to motivate the contributions of this thesis.

Figure 2.1 – Teleoperation is an effective way to collect data for robot learning (Zhang et al., 2018).

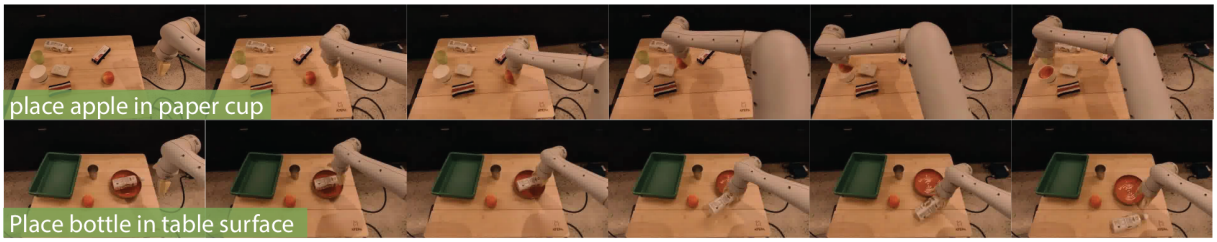


Figure 2.2 – Behavior cloning at scale can learn to generalize to new tasks and new settings (Jang et al., 2022).

2.1 Learning control policies

In classical robotics, movements of robots are traditionally achieved through the application of techniques rooted in optimal control methods (Bellman, 1966; Raibert, 1986; Koenemann et al., 2015), which primarily require a mathematical model of the robot to generate the desired motions. One objective of robot learning is instead to employ data-driven learning methods to acquire control policies directly from interaction data. In the following, we provide an overview of various approaches for learning robot policies in this manner.

2.1.1 Imitation Learning

Imitation learning involves training a policy to replicate the behaviors demonstrated by an expert. In this scenario, we are provided with robot demonstrations collected by expert operators, and the objective is to train deep neural network policies to accurately reproduce these behaviors. These demonstrations can be gathered through either teleoperation (Zhang et al., 2018), where a human operator controls the robot to record the demonstrations as illustrated in Figure 2.1, or by designing scripts to streamline the process of collecting these demonstrations. In any case, substantial human efforts are required for both of these approaches.

Behavior cloning (Pomerleau, 1988) is perhaps the simplest way to learn policies for robotic control: given robot demonstrations, the policy learns to regress actions conditioned on robot observations with supervised learning (Bojarski et al., 2017; Chen and Huang, 2017). Behavior cloning is an intriguing approach due to the stability offered by supervised learning when training deep neural networks. When applied to a wide range of demonstrations spanning various tasks and environments, this method has the potential to achieve effective generalization, allowing it to perform well on new tasks and unseen settings (Jang et al., 2022; Brohan et al., 2022) (Figure 2.2).

In continuous control, behavior cloning is commonly used to train policies by minimizing the mean squared error between the predicted actions generated by the policy and

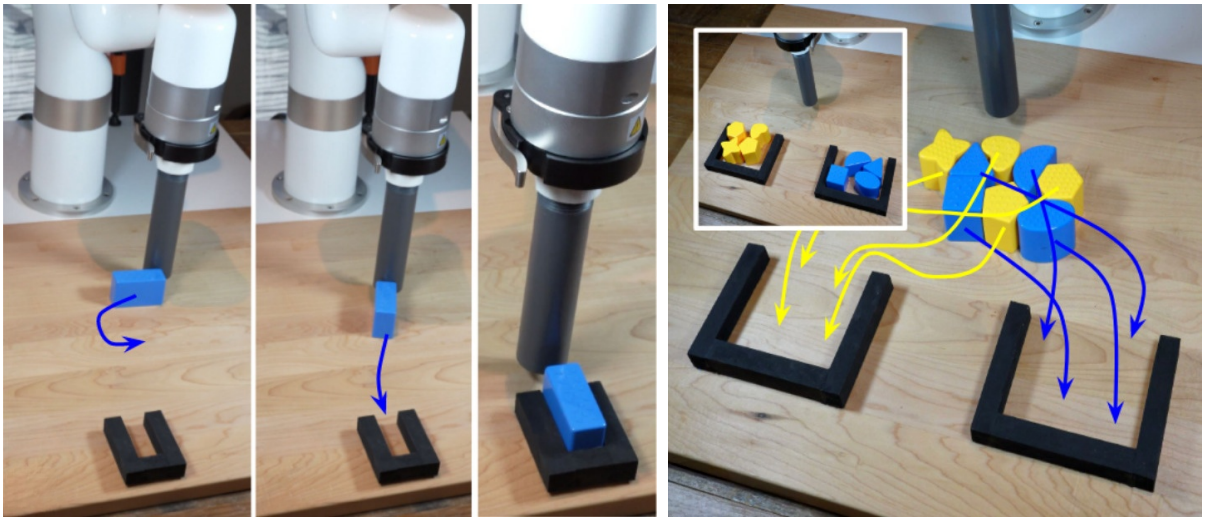


Figure 2.3 – Multimodal policies trained with imitation learning, such as Implicit Behavior cloning (Florence et al., 2022) can solve precise manipulation tasks: (left) precise oriented insertion requiring 1mm precision and (right) combinatorially complex sorting task.

the expert actions. However, a challenge arises when dealing with multi-modal demonstrations, which is often the case when working with data collected by human operators. In such scenarios, fitting a unimodal policy becomes challenging. To overcome this limitation, some works have proposed to use more complex multi-modal policies (Shafiullah et al., 2022; Chi et al., 2023; Zhao et al., 2023; Florence et al., 2022). These policies aim to address precise manipulation tasks with fewer demonstrations, offering enhanced capabilities and adaptability, as illustrated in Figure 2.3

When expert data coverage is not sufficient, the policy errors may accumulate over time, progressively leading the agent to states not covered by the expert data. To mitigate this compounding errors problem (Xu et al., 2020), other work have extended imitation learning beyond behavior cloning as a divergence minimization between the robot policy and the expert state-action distribution (Ho and Ermon, 2016; Ke et al., 2021; Ghasemipour et al., 2020). One notable way is through inverse reinforcement learning (Ng et al., 2000; Abbeel and Ng, 2004; Aghasadeghi and Bretl, 2011; Fu et al., 2017), which involves estimating a reward function that characterizes demonstrations as behaviors that are nearly optimal, and training a policy to maximize this reward function.

2.1.2 Reinforcement Learning

An alternative approach to imitation learning is to define the task through a scalar reward function and to learn a policy to maximize it using reinforcement learning (Sutton and Barto, 2018). In this scenario, expert demonstrations are not required, as the policy can autonomously collect its own training data online. Recent advancements in employ-

Figure 2.4 – Rudin et al. (2022) learns locomotion policies with reinforcement learning in massively paralleled simulations (top) that can transfers to real quadruped robots (bottom).

ing deep neural networks as function approximators for policy learning have facilitated the scalability of deep reinforcement learning to handle increasingly complex sequential decision-making problems, which may involve high-dimensional states and actions (Mnih et al., 2015; Silver et al., 2016). However, applying deep reinforcement learning to robotics introduces additional hurdles, as it is impractical and potentially unsafe to collect online data directly from real robots (Pinto and Gupta, 2016; Kalashnikov et al., 2018; Levine et al., 2018). Consequently, a significant emphasis in the advancement of deep reinforcement learning for robotics has been placed on designing sample-efficient algorithms (Nagabandi et al., 2020; Smith et al., 2022; Wu et al., 2023) as well as training policies in simulated environments before transferring them to real robots (Tobin et al., 2017; Peng et al., 2018a; Akkaya et al., 2019; Lee et al., 2020; Rudin et al., 2022), as illustrated in Figure 2.4.

Many approaches cast reinforcement learning as a probabilistic inference problem where the optimal policy should match a probability distribution in a graphical model defined by the reward and the environment dynamics (Toussaint, 2009; Kappen et al., 2012; Levine, 2018). This formulation makes apparent the existence of a prior policy (Haarnoja et al., 2017; Abdolmaleki et al., 2018b; Haarnoja et al., 2018b) which can be adapted to incur additional knowledge to the policy (Teh et al., 2017; Galashov et al., 2019; Tirumala et al., 2019).

Another set of works aims to solve reinforcement learning through outcome-conditioned action regression (Lynch et al., 2020; Emmons et al., 2021; Furuta et al., 2021), where policies are trained via behavior cloning conditioned on future outcome of the full trajectory. Such future outcome corresponds to trajectory information such as future returns (Kumar et al., 2019b; Schmidhuber, 2019; Srivastava et al., 2019), many previous timesteps (Chen et al., 2021b; Janner et al., 2021) or a desired goal-configuration (Ding et al., 2019; Ghosh et al., 2019; Lynch et al., 2020; Emmons et al., 2021). Consequently, these policies exhibit multitask capabilities, leveraging the generalization abilities of deep neural networks to achieve rewards surpassing those in the dataset or to attain previously unseen goals.

Traditional reinforcement learning typically focuses on maximizing rewards by iteratively switching between online data collection in the environment and enhancing the learned policy. In contrast, offline reinforcement learning (Levine et al., 2020) aims to learn policies from a fixed dataset of interactions, without the ability to interact with the environment during training (Fujimoto et al., 2018; Kumar et al., 2019a; Nair et al., 2020; Kumar et al., 2020; Kostrikov et al., 2021). This is particular relevant when it comes to directly learning policies on real robots (Singh et al., 2020; Chebotar et al., 2021; Kumar et al., 2022b), as it is impractical to collect a new dataset for every experiment. Even when expert demonstrations are available, in the presence of reward functions, it is often advantageous to choose offline reinforcement learning as a preferred approach rather than relying solely on behavior cloning (Kumar et al., 2022a).

2.1.3 Hierarchical policies

Policies trained with imitation learning and reinforcement learning tend to struggle with temporally extended reasoning. Long-horizon tasks can be addressed by hierarchical reinforcement learning (Dayan and Hinton, 1993; Wiering and Schmidhuber, 1997; Dietterich, 2000; Levy et al., 2019; Vezhnevets et al., 2017). Such methods often design high-level policies that operate at a coarser time scale and control execution of low-level policies. Hierarchical reinforcement learning approaches are generally based either on options (Sutton et al., 1999) or feudal framework (Dayan and Hinton, 1993). Option methods involve learning a higher-level policy that switches between individual skill policies (Bacon et al., 2017; Frans et al., 2017; Lee et al., 2019; Florensa et al., 2017; Strudel et al., 2020), while feudal approaches involve learning a higher-level policy that modulates a lower-level policy by a control signal (Nachum et al., 2018; Haarnoja et al., 2018a; Vezhnevets et al., 2017; Kulkarni et al., 2016; Hausman et al., 2018).

In our work, we delved further into goal-conditioned reinforcement learning (Kaelbling, 1993; Andrychowicz et al., 2017; Pong et al., 2018; Nair et al., 2018; Eysenbach et al., 2020a), where the policy and the reward are conditioned on a desired goal to be achieved

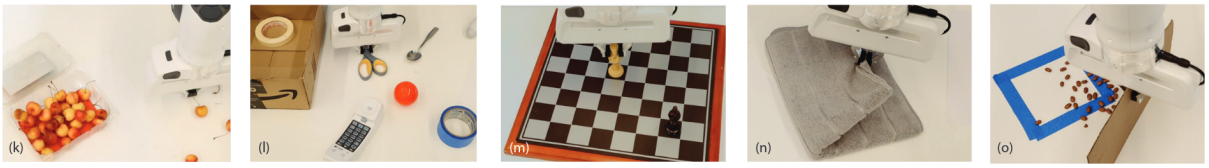


Figure 2.5 – CLIPort (Shridhar et al., 2022) uses pretrained CLIP (Radford et al., 2021) image and text encoders to perfrom language-conditioned visual manipulation tasks.

in the environment. This approach offers opportunities for effectively communicating tasks to robots and enables hierarchical reasoning by leveraging the compositional nature of goal-conditioned tasks. Indeed, high-level policies can be learned to iteratively predict a sequence of intermediate subgoals. Such subgoals can then be used to modulate low-level policies (Nachum et al., 2018; Gupta et al., 2019a; Nair and Finn, 2020). As an alternative to the iterative planning, other methods generate sequences of subgoals with a divide-and-conquer approach (Jurgenson et al., 2020; Parascandolo et al., 2020; Pertsch et al., 2020b). In contrast to previous works, in this thesis we showcase the effectiveness of utilizing a high-level policy to guide policy learning during training. This approach enables us to successfully tackle temporally extended tasks without the need for planning subgoals at inference.

Recently, large language models (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; Huang et al., 2022) have showcased their capability to offer high-level planning by generating sequences of language instructions. These instructions can then be followed by low-level language-conditioned policies to effectively tackle long-horizon tasks (Ahn et al., 2022; Liang et al., 2022; Driess et al., 2023; Shah et al., 2023; Singh et al., 2023).

2.2 Robot learning with human data

One of the major difficulties in learning control policies for robots stems from the absence of large and diverse datasets of robot demonstrations. One possible approach to mitigate this challenge is by incorporating additional sources of data that were not specifically collected for robots into the process of robot learning.

2.2.1 Pretraining for robotics

One strategy to incorporate human data into robot learning is by incorporating pre-trained models, trained on human data, into the robot policies (Figure 2.5) By integrating pretrained policy components derived from large and diverse datasets, the resulting robot policies are expected to exhibit greater ability for generalization.

Notably, the common approach to enable robot policies to understand language instructions is to use a frozen text encoder, pretrained on text data unrelated to robotics, to encode the given text instructions (Lynch and Sermanet, 2020; Jang et al., 2022; Shridhar et al., 2022; Guhur et al., 2023). When training with sufficiently diverse captioned robot demonstrations, this approach exhibits the capability to generalize to unseen instructions.

As for pretrained image representations, object-centric approaches focus on estimating the poses of known objects in the scene (Rothganger et al., 2006; Liu et al., 2021; Rad and Lepetit, 2017; Wang et al., 2019; Labbé et al., 2020, 2022), then plan and manipulate the object according to the recovered poses (Choi et al., 2012; Litvak et al., 2019; Stevšić et al., 2020; Chabal et al., 2022). However, pose estimation for control typically requires access to known 3D models of the objects to manipulate and may overlook crucial control-related information by solely focusing on object poses.

Leveraging vast datasets like web images to learn broader representations has proven to be highly successful for various downstream computer vision tasks (Deng et al., 2009; Radford et al., 2021; Chen et al., 2020; He et al., 2022; Bardes et al., 2021). However, these kind of representations don't always transfer well to visual motor control (Parisi et al., 2022). Recently, as an alternative to train control representations from scratch (Laskin et al., 2020; Kostrikov et al., 2020), there has been a growing interest in learning a similar general image representations for motor control by training on large and diverse datasets of human videos (Nair et al., 2022b; Xiao et al., 2022; Ma et al., 2022; Radosavovic et al., 2023; Karamcheti et al., 2023; Ma et al., 2023). Several studies have explored joint vision-language pretraining for tasks that involve language conditioning (Shridhar et al., 2022; Liu et al., 2022). Nonetheless, how we can obtain a general visual representation for robotics is still an open question to this day (Majumdar et al., 2023; Sharma et al., 2023). Similarly, our study in this thesis reveals that video encoders trained for action recognition can serve as effective skill representations for visual manipulation tasks.

2.2.2 Imitating human videos

Pretraining holds the potential to unlock more generalizable policies in robotics by improving the representations of states or instructions. Yet, it does not directly resolve the issue of insufficient robot behavior demonstrations. One approach to tackle this challenge involves leveraging human videos for behavior imitation, allowing the acquisition of robot behaviors through observational learning. However, bridging the embodiment gap between human videos and the robot poses a significant challenge.

Many previous works have hence revolved around imitating a single video featuring a human, or even an animal (Figure 2.6), showcasing the target task. To accomplish this, one approach is to extract keypoints or human poses from the instructional videos and

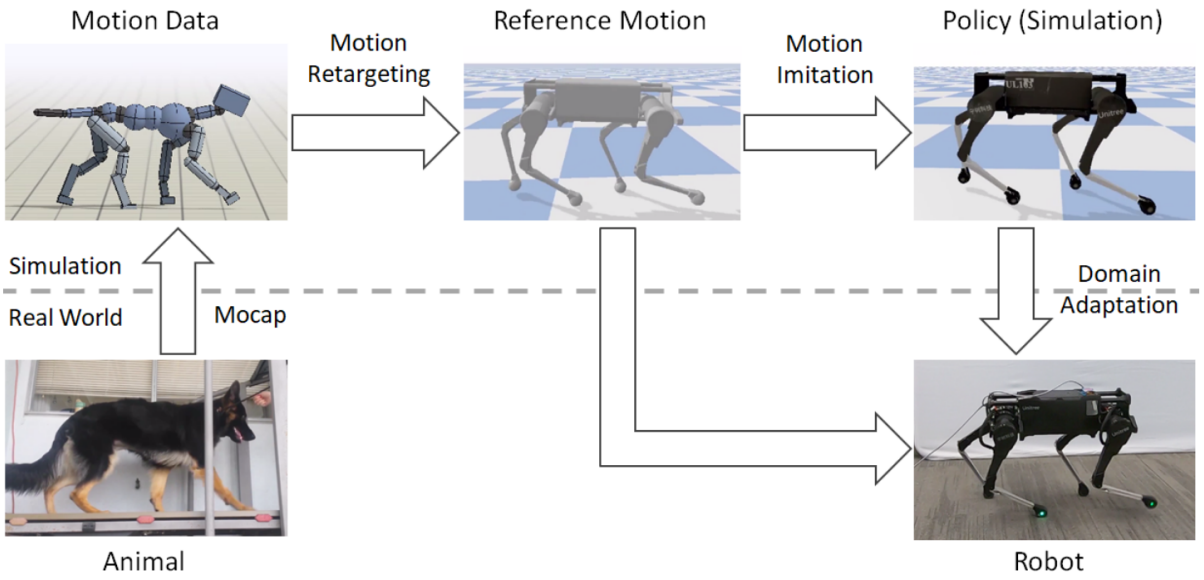


Figure 2.6 – Peng et al. (2020) put forward a methodology to acquire quadruped locomotion skills by watching animals.

subsequently replicate them within the target robot environment, thereby imitating the demonstrated actions (Peng et al., 2018b; Das et al., 2020; Peng et al., 2020; Xiong et al., 2021; Zorina et al., 2021). An alternative strategy involves transforming the human video into a robot video by employing image translation or inpainting techniques to substitute the human body with a robot representation, enabling its execution thereafter (Liu et al., 2018; Sharma et al., 2019; Smith et al., 2019; Bahl et al., 2022). Nonetheless, these approaches are confined to a single task and often necessitate meticulous alignment between the video and robot settings in advance, such as collecting the video in controlled lab environments or modifying the robot environment to resemble the video setting as closely as possible. These constraints hinder the ability to leverage large-scale sources of video data for robot learning.

More recently, people have been considering how we could leverage behaviors from a large number of videos for robotics (Sermanet et al., 2018; Petrik et al., 2020). Other works have shown that we can infer states and actions from diverse videos and use it for reinforcement learning (Edwards and Isbell, 2019; Schmeckpeper et al., 2020a,b; Seo et al., 2022; Baker et al., 2022). Prior works have also considered leveraging large datasets of human videos to learn reward functions for robotics manipulations (Shao et al., 2021; Chen et al., 2021a; Ma et al., 2022; Alakuijala et al., 2023). Likewise, our research in this thesis endeavors to propose methods for following human video demonstrations by learning a video similarity on large datasets of annotated human videos.

Chapter 3

Goal-Conditioned Reinforcement Learning with Imagined Subgoals

In this chapter, a task is specified by the desired goal configuration in the environment of the robot agent. We propose a novel reinforcement learning algorithm designed to enhance goal-conditioned policies with temporal reasoning capabilities.

Goal-conditioned reinforcement learning endows an agent with a large variety of skills, but it often struggles to solve tasks that require more temporally extended reasoning. In this work, we propose to incorporate imagined subgoals into policy learning to facilitate learning of complex tasks. Imagined subgoals are predicted by a separate high-level policy, which is trained simultaneously with the policy and its critic. This high-level policy predicts intermediate states halfway to the goal using the value function as a reachability metric. We don't require the policy to reach these subgoals explicitly. Instead, we use them to define a prior policy, and incorporate this prior into a KL-constrained policy iteration scheme to speed up and regularize learning. Imagined subgoals are used during policy learning, but not during test time, where we only apply the learned policy.

We evaluate our approach on complex robotic navigation and manipulation tasks, and demonstrate significant improvements compared to previous methods. We also discuss and ablate several design choices of our method.

3.1 Introduction

An intelligent agent aims at solving tasks of varying horizons in an environment. It must be able to identify how the different tasks intertwine with each other and leverage behaviors learned by solving simpler tasks to efficiently master more complex tasks. For instance, once a legged robot has learned how to walk in every direction in a simulated

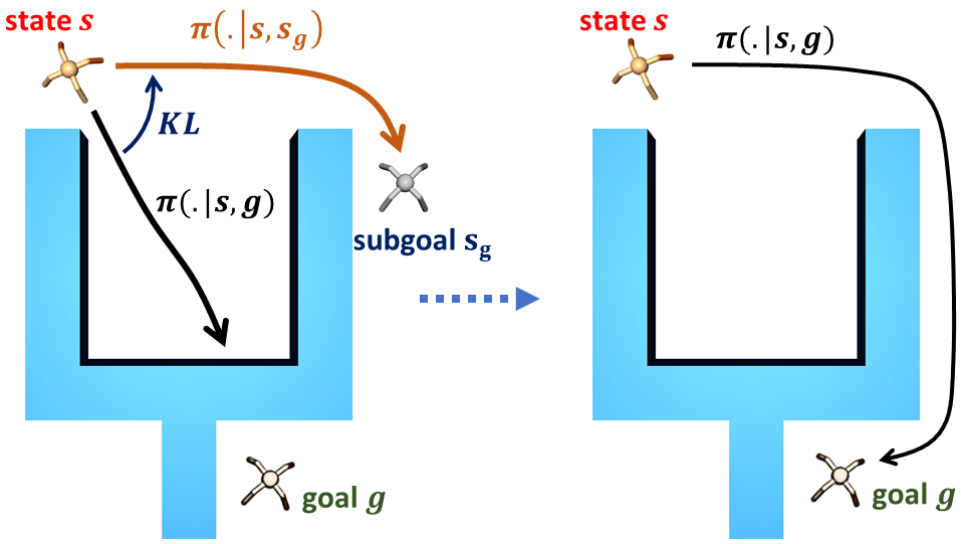


Figure 3.1 – Illustration of the KL-regularized policy learning using imagined subgoals. (Left): The policy fails to reach a distant goal, yet it can reach a closer subgoal. Our approach automatically generates imagined subgoals for a task and uses such subgoals to direct the policy search during training. (Right): At test time, the resulting flat policy can reach arbitrarily distant goals *without* relying on subgoals.

maze environment, it could use these behaviors to efficiently learn to navigate in this environment. Goal-conditioned reinforcement learning (RL) (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017) defines each task by the desired goal and, in principle, could learn a wide range of skills. In practice, however, reinforcement learning struggles to perform temporally extended reasoning.

Hierarchical methods have proven effective for learning temporal abstraction in long-horizon problems (Dayan and Hinton, 1993; Wiering and Schmidhuber, 1997; Sutton et al., 1999; Dietterich, 2000). In the goal-conditioned setting, a high-level controller typically finds an appropriate sequence of subgoals that can be more easily followed by a low-level policy (Nachum et al., 2018; Gupta et al., 2019a). When chosen appropriately, these subgoals effectively decompose a complex task into easier tasks. However, hierarchical methods are often unstable to train (Nachum et al., 2018) and rely on appropriate temporal design choices.

In this work, we propose to use subgoals to improve the goal-conditioned policy. Instead of reaching the subgoals explicitly, our method builds on the following intuition. If the current policy can readily reach a subgoal, it could provide guidance for reaching more distant goals, as illustrated in Figure 3.1. We apply this idea to all possible state and goal pairs of the environment. This self-supervised approach progressively extends the horizon of the agent throughout training. At the end of training, the resulting flat policy does not require access to subgoals and can reach distant goals in its environment.

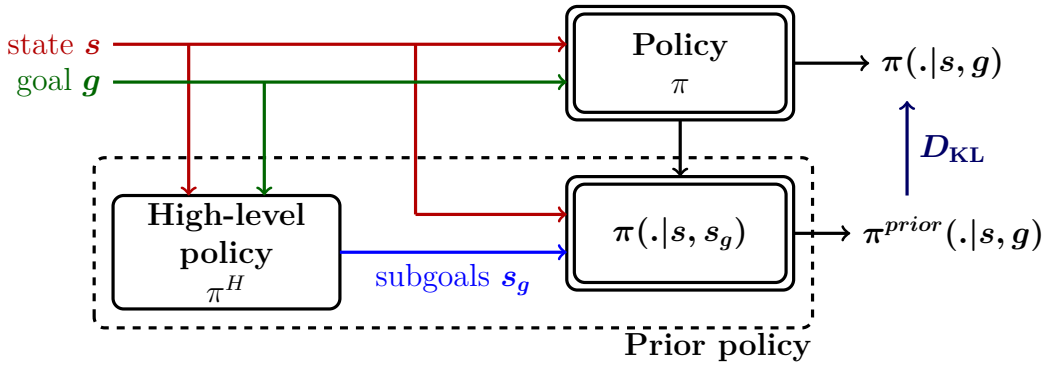


Figure 3.2 – Overview over our Reinforcement learning with Imagined Subgoals (RIS) approach. During policy training, the policy π is constrained to stay close to the prior policy π^{prior} through KL-regularization. We define the prior policy π^{prior} as the distribution of actions required to reach intermediate subgoals s_g of the task. Given the initial state s and the goal state g, the subgoals are generated by the high-level policy π^H. Note that the high-level policy and subgoals are only used during training of the target policy π. At test time we directly use π to generate appropriate actions.

Our method does not require a full sequence of subgoals, but predicts subgoals that are halfway to the goal, such that reaching them corresponds to a lower level of temporal abstraction than reaching the goal. To handle this subgoal prediction problem, we simultaneously train a separate high-level policy operating in the state space and use the value function of the goal-conditioned policy as a relative measure of distance between states (Eysenbach et al., 2019; Nasiriany et al., 2019).

To incorporate subgoals into policy learning, we introduce a prior policy defined as the distribution over actions required by the policy to reach intermediate subgoals (Figure 3.2). As we will show, this distribution can also be implicitly defined in terms of the action-value function. When using appropriate subgoals that are easier to reach, this prior policy corresponds to an initial guess to reach the goal. Accordingly, we leverage a policy iteration with an additional Kullback-Leibler (KL) constraint scheme towards this prior policy. This adequately encourages the policy to adapt the simpler behaviors associated with reaching these subgoals to master more complex goal-reaching tasks in the environment. Because these subgoals are *not* actively pursued when interacting with the actual environment but only used to accelerate policy learning, we refer to them as *imagined subgoals*. Imagined subgoals and intermediate subgoals have equal meaning in this paper.

Our method, Reinforcement learning with Imagined Subgoals (RIS), builds upon off-policy actor-critic approaches for continuous control and additionally learns a high-level policy that updates its predictions according to the current goal-reaching capabilities of the policy. Our approach is general, solves temporally extended goal-reaching tasks with sparse rewards, and self-supervises its training by choosing appropriate imagined subgoals

to accelerate its learning. Furthermore, we extend our approach to vision-based environments, where imagined subgoals are predicted in the feature space of the learned image encoder. We highlight key design choices in this setting and demonstrate improvements on a vision-based robotic manipulation task.

In summary, our contributions are threefold: (1) we propose a method for predicting subgoals which decomposes a goal-conditioned task into easier subtasks; (2) we incorporate these subgoals into policy learning through a KL-regularized policy iteration scheme with a specific choice of prior policy; and (3) we show that our approach greatly accelerates policy learning on a set of simulated robotics environments that involve motor control and temporally extended reasoning. [1]

3.2 Related Work

Goal-conditioned reinforcement learning has been addressed by a number of methods (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017; Veeriah et al., 2018; Pong et al., 2018; Nair et al., 2018; Zhao et al., 2019; Pitis et al., 2020; Eysenbach et al., 2020a). Given the current state and the goal, the resulting policies predict action sequences that lead towards the desired goal. Hindsight experience replay (HER) (Kaelbling, 1993; Andrychowicz et al., 2017) is often used to improve the robustness and sample efficiency of goal-reaching policies. While in theory such policies can address any goal-reaching task, they often fail to solve temporally extended problems in practice (Levy et al., 2019; Nachum et al., 2018).

Long-horizon tasks can be addressed by hierarchical reinforcement learning (Dayan and Hinton, 1993; Wiering and Schmidhuber, 1997; Dietterich, 2000; Levy et al., 2019; Vezhnevets et al., 2017). Such methods often design high-level policies that operate at a coarser time scale and control execution of low-level policies. To address goal-conditioned settings, high-level policies can be learned to iteratively predict a sequence of intermediate subgoals. Such subgoals can then be used as targets for low-level policies (Nachum et al., 2018; Gupta et al., 2019a; Nair and Finn, 2020). As an alternative to the iterative planning, other methods generate sequences of subgoals with a divide-and-conquer approach (Jurgenson et al., 2020; Parascandolo et al., 2020; Pertsch et al., 2020b). While hierarchical RL methods can better address long-horizon tasks, the joint learning of high-level and low-level policies may imply instabilities (Nachum et al., 2018). Similar to previous hierarchical RL methods, we use subgoals to decompose long-horizon tasks into simpler problems. Our subgoals, however, are only used during policy learning to guide and accelerate the search of the non-hierarchical policy.

1. Code is available on the project webpage `https://www.di.ens.fr/willow/research/ris/`.

Several recent RL methods use the value function of goal-reaching policies to measure distances between states and to plan sequences of subgoals (Nasiriany et al., 2019; Eysenbach et al., 2019; Zhang et al., 2020). In particular, LEAP (Nasiriany et al., 2019) uses the value function of TDM policies (Pong et al., 2018) and optimizes sequences of appropriate subgoals at test time. Similar to previous methods, we use the value function as a distance measure between states. Our method, however, avoids expensive test-time optimization of subgolas. We also experimentally compare our method with LEAP and demonstrate improvements. Moreover, we show that our approach can benefit from recent advances in learning representations for reinforcement learning from pixels on a vision-based robotic manipulation task (Kostrikov et al., 2020; Laskin et al., 2020). In a parallel work to ours, Zhang et al. (2021) use goal-conditioned value functions for the search of subgoals and propose to use subgoals only when collecting new experience and not during testing.

Many approaches cast reinforcement learning as a probabilistic inference problem where the optimal policy should match a probability distribution in a graphical model defined by the reward and the environment dynamics (Toussaint, 2009; Kappen et al., 2012; Levine, 2018). Several methods optimize an objective incorporating the divergence between the target policy and a prior policy. The prior policy can be fixed (Haarnoja et al., 2017; Abdolmaleki et al., 2018b; Haarnoja et al., 2018b; Pertsch et al., 2020a) or learned jointly with the policy (Teh et al., 2017; Galashov et al., 2019; Tirumala et al., 2019). While previous work imposes explicit priors e.g., in the multi-task and transfer learning settings (Teh et al., 2017; Galashov et al., 2019; Tirumala et al., 2019), we constrain our policy search by the prior distribution implicitly defined by subgoals produced by the high-level policy.

Behavior priors have often been used to avoid value overestimation for out-of-distribution actions in offline reinforcement learning (Fujimoto et al., 2018; Wu et al., 2019; Kumar et al., 2019a; Siegel et al., 2020; Nair et al., 2020; Wang et al., 2020). Similar to these methods, we constrain our high-level policy to avoid predicting subgoals outside of the valid state distribution.

Finally, our work shares similarities with guided policy search methods (Levine and Koltun, 2013; Levine and Abbeel, 2014; Levine et al., 2016) which alternate between generating expert trajectories using trajectory optimization and improving the learned policy. In contrast, our policy search is guided by subgoals produced by a high-level policy.

3.3 Method

Our method builds on the following key observation. If an action a is well-suited for approaching an intermediate subgoal s_g from state s, it should also be a good choice for approaching the final goal g of the same task. We assume that reaching subgoals s_g is simpler than reaching more distant goals g. Hence, we adopt a self-supervised strategy and use the subgoal-reaching policy $\pi(\cdot|s, s_g)$ as a guidance when learning the goal-reaching policy $\pi(\cdot|s, g)$. We denote our approach as Reinforcement learning with Imagined Subgoals (RIS) and present its overview in Figures 3.1 and 3.2.

To implement the idea of RIS, we first introduce a high-level policy π^H predicting imagined subgoals s_g, as described in Section 3.3.2. Section 3.3.3 presents the regularized learning of the target policy π using subgoals. The joint training of π and π^H is summarized in Section 3.3.4. Before describing our technical contributions, we present the actor-critic paradigm used by RIS in Section 3.3.1 below.

3.3.1 Goal-Conditioned Actor-Critic

We consider a discounted, infinite-horizon, goal-conditioned Markov decision process, with states $s \in \mathcal{S}$, goals $g \in \mathcal{G}$, actions $a \in \mathcal{A}$, reward function $r(s, a, g)$, dynamics $p(s'|s, a)$ and discount factor γ. The objective of a goal-conditioned RL agent is to maximize the expected discounted return

$$J(\pi) = \mathbb{E}_{g\sim\rho_g,\tau\sim d^\pi(.|g)}[\sum_t \gamma^t r(s_t, a_t, g)]$$

under the distribution

$$d^\pi(\tau|g) = \rho_0(s_0) \prod_t \pi(a_t|s_t, g)p(s_{t+1}|s_t, a_t)$$

induced by the policy π and the initial state and goal distribution. The policy $\pi(.|s, g)$ in this work generates a distribution over continuous actions a conditioned on state s and goal g. Many algorithms rely on the appropriate learning of the goal-conditioned action-value function Q^π and the value function V^π defined as

$$Q^\pi(s, a, g) = \mathbb{E}_{s_0=s,a_0=a,\tau\sim d^\pi(.|g)}[\sum_t \gamma^t r(s_t, a_t, g)]$$

and

$$V^\pi(s, g) = \mathbb{E}_{a\sim\pi(.|s,g)} Q^\pi(s, a, g).$$

In this work we assume states and goals to co-exist in the same space, i.e. $\mathcal{S} = \mathcal{G}$,

where each state can be considered as a potential goal. Moreover, we set the reward r to -1 for all actions until the policy reaches the goal.

We follow the standard off-policy actor-critic paradigm (Silver et al., 2014; Heess et al., 2015; Mnih et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018b). Experience, consisting of single transition tuples (s_t, a_t, s_{t+1}, g), is collected by the policy in a replay buffer D. Actor-critic algorithms maximize return by alternating between policy evaluation and policy improvement. During the policy evaluation phase, a critic $Q^\pi(s, a, g)$ estimates the action-value function of the current policy π by minimizing the Bellman error with respect to the Q-function parameters ϕ_k:

$$Q_{\phi_{k+1}} = \arg\min_{\phi} \frac{1}{2}\mathbb{E}_{(s_t,a_t,s_{t+1},g)\sim D}[(y_t - Q_\phi(s_t, a_t, g))^2] \tag{3.1}$$

with the target value

$$y_t = r(s_t, a_t, g) + \gamma \mathbb{E}_{a_{t+1}\sim\pi(.|s_{t+1},g)} Q_{\phi_k}(s_{t+1}, a_{t+1}, g).$$

During the policy improvement phase, the actor π is typically updated such that the expected value of the current Q-function Q^π, or alternatively the advantage $A^\pi(s, a, g) = Q^\pi(s, a, g) - V^\pi(s, g)$, under π is maximized:

$$\pi_{\theta_{k+1}} = \arg\max_{\theta} \mathbb{E}_{(s,g)\sim D, a\sim\pi_\theta(.|s,g)}[Q^\pi(s, a, g)]. \tag{3.2}$$

Reaching distant goals using delayed rewards may require expensive policy search. To accelerate the training, we propose to direct the policy search towards intermediate subgoals of a task. Our method learns to predict appropriate subgoals by the high-level policy π^H. The high-level policy is trained together with the policy π as explained below.

3.3.2 High-Level Policy

We would like to learn a high-level policy $\pi^H(.|s, g)$ that predicts an appropriate distribution of imagined subgoals s_g conditioned on valid states s and goals g. Our high-level policy is defined in terms of the policy π and relies on the goal-reaching capabilities of π as described next.

Subgoal search with a value function. We note that our choice of the reward function $r = -1$ implies that the norm of the value function $V^\pi(s, g)$ corresponds to an estimate of the expected discounted number of steps required for the policy to reach the goal g from the current state s. We therefore propose to use $|V^\pi(s^i, s^j)|$ as a measure of the distance between any valid states s^i and s^j. Note that this measure depends on the

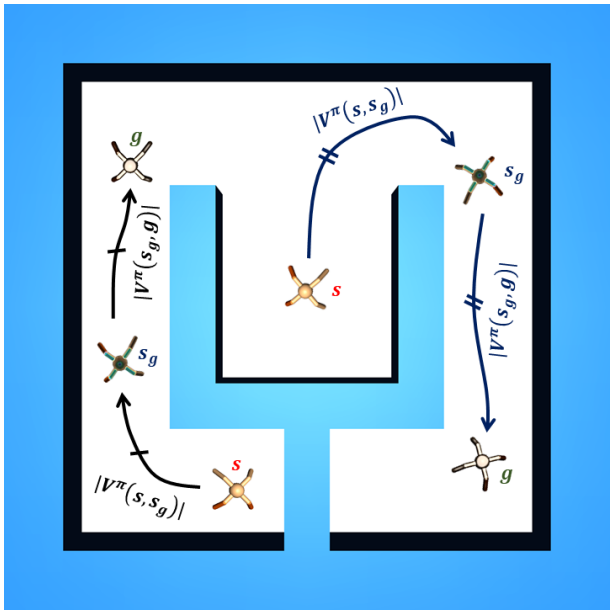


Figure 3.3 – Given initial states s and goal states g, our subgoals are states on the middle of the path from s to g. We measure distances between states by the value function $|V^\pi(s_1, s_2)|$ corresponding to the current policy π. We obtain a distribution of subgoals from the high-level policy $s_g \sim \pi^H(.|s, g)$. We use subgoals *only* at the training to regularize and accelerate the policy search.

policy π and evolves with the improvements of π during training.

Reaching imagined subgoals of a task should be easier than reaching the final goal. To leverage this assumption for policy learning, we need to find appropriate subgoals. In this work we propose to define subgoals s_g as midpoints on the path from the current state s to the goal g, see Figure 3.3. More formally, we wish s_g (i) to have equal distance from s and g, and (ii) to minimize the length of the paths from s to s_g and from s_g to g. We can find subgoals that satisfy these constraints by using our distance measure $|V^\pi(s^i, s^j)|$ and minimizing the following cost $C_\pi(s_g|s, g)$

$$C_\pi(s_g|s, g) = \max\left(|V^\pi(s, s_g)|, |V^\pi(s_g, g)|\right). \tag{3.3}$$

However, naively minimizing the cost C_π under the high-level policy distribution, i.e.

$$\pi^H_{k+1} = \arg\min_{\pi^H} \mathbb{E}_{(s,g)\sim D, s_g\sim\pi^H(.|s,g)}[C_\pi(s_g|s, g)], \tag{3.4}$$

may lead to undesired solutions where the high-level policy samples subgoals outside the valid state distribution $p_s(.)$. Such predictions may, for example, correspond to unfeasible robot poses, unrealistic images, or other adversarial states which have low distance from s and g according to $|V^\pi|$ but are unreachable in practice.

Predicting valid subgoals. To avoid non-valid subgoals, we can additionally encourage the high-level policy to stay close to the valid state distribution $p_s(.)$ using the following KL-regularized objective:

$$\begin{aligned} \pi^H_{k+1} &= \arg\max_{\pi^H} \mathbb{E}_{(s,g)\sim D, s_g\sim\pi^H(.|s,g)} \left[A^{\pi^H_k}(s_g|s,g) \right] \\ &\text{s.t. } D_{\text{KL}}\left(\pi^H(.|s,g) \,||\, p_s(.) \right) \leq \epsilon, \end{aligned} \tag{3.5}$$

where the advantage

$$A^{\pi^H_k}(s_g|s,g) = \mathbb{E}_{\hat{s_g}\sim\pi^H_k(.|s,g)} \left[C_\pi(\hat{s_g}|s,g) \right] - C_\pi(s_g|s,g)$$

quantifies the quality of a subgoal s_g against the high-level policy distribution. The Kullback-Leibler divergence term in (3.5) requires an esimate of the density $p_s(.)$. While estimating the unknown $p_s(.)$ might be challenging, we can obtain samples from this distribution, for example, by randomly sampling states from the replay buffer $s_g \sim D$.

We propose instead to *implicitly* enforce the KL constraint (3.5). We first note that the analytic solution to (3.5) can be obtained by enforcing the Karush–Kuhn–Tucker conditions, for which the Lagrangian is:

$$\mathcal{L}(\pi^H, \lambda) = \mathbb{E}_{s_g\sim\pi(.|s,g)} \left[A^{\pi^H_k}(s_g|s,g) \right] + \lambda \left(\epsilon - D_{\text{KL}}\left(\pi(.|s,g) \,||\, p_s(.) \right) \right).$$

The closed form solution to this problem is then given by:

$$\pi^H_\star(s_g|s,g) = \frac{1}{Z(s,g)} p_s(s_g) \exp\left(\frac{1}{\lambda} A^{\pi^H_k}(s_g|s,g) \right) \tag{3.6}$$

with the normalizing partition function $Z(s,g) = \int p_s(s_g) \exp\left(\frac{1}{\lambda} A^{\pi^H_k}(s_g|s,g) \right) ds_g$ (Peters et al., 2010; Rawlik et al., 2012; Abdolmaleki et al., 2018b,a; Nair et al., 2020). We project this solution into the policy space by minimizing the forward KL divergence between our parametric high-level policy π^H_ψ and the optimal non parametric solution $\pi^H_\star$:

$$\begin{aligned} \pi^H_{\psi_{k+1}} &= \arg\min_\psi \mathbb{E}_{(s,g)\sim D} D_{\text{KL}}\left(\pi^H_\star(.|s,g) \,||\, \pi^H_\psi(.|s,g) \right) \\ &= \arg\max_\psi \mathbb{E}_{(s,g)\sim D, s_g\sim D} \left[\log \pi^H_\psi(s_g|s,g) \frac{1}{Z(s,g)} \exp\left(\frac{1}{\lambda} A^{\pi^H_k}(s_g|s,g) \right) \right], \end{aligned} \tag{3.7}$$

where λ is a hyperparameter. Conveniently, this policy improvement step corresponds to a weighted maximum likelihood with subgoal candidates obtained from the replay buffer randomly sampled among states visited by the agent in previous episodes. The samples are re-weighted by their corresponding advantage, implicitly constraining the high-level policy to stay close to the valid distribution of states.

Alternatively, we can leverage advantage filtering to train the high-level policy:

$$\pi^H_{\psi_{k+1}} = \arg\max_{\psi} \mathbb{E}_{(s,g)\sim D, s_g \sim D}\left[\log \pi^H_\psi(s_g|s,g)\mathbb{1}_{A^{\pi^H_k}(s_g|s,g)\geq 0}\right], \tag{3.8}$$

which corresponds to training the high-level policy with behavior cloning on subgoal candidates $s_g \sim D$, but only when $s_g \sim D$ has a higher-value than $s_g \sim \pi^H(.|s,g)$. Using the goal-conditioned value function as a comparison metric enables to filter out subgoals that would lead to the worse performance compared to subgoals predicted by the current high-level policy.

3.3.3 Policy Improvement with Imagined Subgoals

Our method builds on the following key insight. If we assume s_g to be an intermediate subgoal on the optimal path from s to g, then the optimal action for reaching g from s should be similar to the optimal action for reaching s_g from s (see Figure 3.1). We can formalize this using a KL constraint on the policy distribution, conditioned on goals g and s_g:

$$D_{\mathrm{KL}}\left(\pi(.|s,g) \,||\, \pi(.|s,s_g)\right) \leq \epsilon.$$

We introduce the non-parametric prior policy $\pi^{prior}(.|s,g)$ as the distribution over actions that would be chosen by the policy π for reaching subgoals $s_g \sim \pi^H(.|s,g)$ provided by the high-level policy. Given a state s and a goal g, we bootstrap the behavior of the policy at subgoals $s_g \sim \pi^H(.|s,g)$ (see Figure 3.2):

$$\pi^{prior}_k(a|s,g) := \mathbb{E}_{s_g\sim\pi^H(.|s,g)}[\pi_{\theta'_k}(a|s,s_g)]. \tag{3.9}$$

As we assume the subgoals to be easier to reach compared to reaching the final goals, this prior policy provides a good initial guess to constrain the policy search to the most promising actions.

Alternatively, we can also define the prior policy in terms of the Boltzmann policy $\pi^{ebm}(.|s,g) :\propto \exp\frac{1}{\beta}Q(s,.,g)$ associated with the learned Q-function instead:

$$\pi^{prior}(.|s,g) :\propto \mathbb{E}_{s_g\sim\pi^H(.|s,g)}\left[\exp\frac{1}{\beta}Q(s,.,s_g)\right]. \tag{3.10}$$

We then propose to leverage a policy iteration scheme with additional KL constraint to shape the policy behavior accordingly. During the policy improvement step, in addition to maximizing the Q-function as in (3.2), we encourage the policy to stay close to the

prior policy through the KL-regularization:

$$\pi_{\theta_{k+1}} = \arg\max_{\theta} \mathbb{E}_{(s,g)\sim D} \mathbb{E}_{a\sim\pi_\theta(.|s,g)} \left[Q^\pi(s,a,g) - \alpha D_{\mathrm{KL}} \left(\pi_\theta(.|s,g) \,||\, \pi_k^{prior}(.|s,g) \right) \right], \tag{3.11}$$

where α is a hyperparameter.

In practice, we found that using an exponential moving average of the online policy parameters to construct the prior policy is necessary to ensure convergence:

$$\theta'_{k+1} = \tau\theta_k + (1-\tau)\theta'_k, \ \tau \in]0,1[. \tag{3.12}$$

This ensures that the prior policy produces a more stable target for regularizing the online policy.

Using the Boltzmann policy for building the prior, we get instead:

$$\begin{aligned} \pi_{\theta_{k+1}} &= \arg\max_{\theta} \mathbb{E}_{(s,g)\sim D} \mathbb{E}_{a\sim\pi_\theta(.|s,g)} \left[Q^\pi(s,a,g) - \alpha D_{\mathrm{KL}} \left(\pi_\theta(.|s,g) \,||\, \pi_k^{prior}(.|s,g) \right) \right] \\ &= \arg\max_{\theta} \mathbb{E}_{(s,g)\sim D} \mathbb{E}_{a\sim\pi_\theta(.|s,g)} \\ &\left[Q^\pi(s,a,g) + \alpha \log \mathbb{E}_{s_g\sim\pi^H(.|s,g)} \exp \frac{1}{\beta} Q(s,a,s_g) - \alpha \log \pi_\theta(a|s,g) \right]. \end{aligned} \tag{3.13}$$

Using this loss, the policy is trained to maximize both the Q-values of the goal and of subgoals predicted by the high-level policy. However in practice, we found that adversarial actions for reaching imagined subgoals that were not regularized by the feedback control loop led to instabilities in some environments. So we constrain the Q values to be in the range of $[0,1]$ by changing the reward function to $r = \mathbb{1}\{\text{goal is reached}\}$ and using a sigmoid output activation function for the Q neural network. Note that with this alternative choice of reward function, the optimal value function is the same as using a reward of -1 until the goal is reached up to an affine transformation. Moreover, when using the Boltzmann prior policy, we use advantage filtering instead of advantage weighting to learn the high-level policy. We experimentally compare these two design choices in Section 3.4.3.5.

3.3.4 Algorithm Summary

We approximate the policy π_θ, the Q-function Q_ϕ and the high-level policy π_ψ^H with neural networks parametrized by θ, ϕ and ψ respectively, and train them jointly using stochastic gradient descent. The Q-function is trained to minimize the Bellman error (3.1), where we use an exponential moving average of the online Q-function parameters to compute the target value. The high-level policy is a probabilistic neu-

Algorithm 1 RL with imagined subgoals

Initialize replay buffer D
Initialize Q_ϕ, π_θ, π_ψ^H
for k = 1, 2, ... **do**
  Collect experience in D using π_θ in the environment
  Sample batch $(s_t, a_t, r_t, s_{t+1}, g) \sim D$ with HER
  Sample batch of subgoal candidates $s_g \sim D$
  Update Q_ϕ using Eq. 3.1 *(Policy Evaluation)*
  Update π_ψ^H Using Eq. 3.7 *(High-Level Policy Improvement)*
  Update π_θ using Eq. 3.11 *(Policy Improvement with Imagined Subgoals)*
end for

ral network whose output parametrizes a Laplace distribution with diagonal variance $\pi_\psi^H(.|s,g) = \text{Laplace}(\mu_\psi^H(s,g), \Sigma_\psi^H(s,g))$ trained to minimize (3.7). The policy is also a probabilistic network parametrizing a squashed Gaussian distribution with diagonal variance $\pi_\psi^H(.|s,g) = \tanh\mathcal{N}(\mu_\theta(s,g), \Sigma_\theta(s,g))$ trained to minimize (3.11). Finally we can approximate π_{prior} using a Monte-Carlo estimate of (3.9).

The policy π and the high-level policy π^H are trained jointly. As the policy learns to reach more and more distant goals, its value function becomes a better estimate for the distance between states. This allows the high-level policy to propose more appropriate subgoals for a larger set of goals. In turn, as the high-level policy improves, imagined subgoals offer a more relevant supervision to shape the behavior of the policy. This virtuous cycle progressively extends the policy horizon further and further away, allowing more complex tasks to be solved by a single policy.

The full algorithm is summarized in Algorithm 1. We use hindsight experience replay (Andrychowicz et al., 2017) (HER) to improve learning from sparse rewards.

3.3.5 Implementation Details

Actor-Critic. Our implementation of the actor-critic algorithm is based on Soft Actor-Critic (Haarnoja et al., 2018c), where we remove the entropy term during policy evaluation and replace the entropy term by the KL divergence between policy and our prior policy during policy improvement. The policy is a neural network that parametrizes the mean and diagonal covariance matrix of a squashed Gaussian distribution

$$\pi_\theta(.|s,g) = \tanh\mathcal{N}(\mu_\theta(s,g), \Sigma_\theta(s,g)).$$

We train two seperate Q-networks with target networks and take the minimum over the two target values to compute the bootstrap value. The target networks are updated using an exponential moving average of the online Q parameters: $\phi'_{k+1} = \tau\phi_k + (1-\tau)\phi'_k$.

High-level policy training. The high-level policy is a neural network that outputs the mean and diagonal covariance matrix of a Laplace distribution

$$\pi_\psi^H(.|s,g) = \text{Laplace}(\mu_\psi(s,g), \Sigma_\psi(s,g)).$$

Following Nair et al. (2020), instead of estimating the normalizing factor $Z(s,g)$ in (3.8), we found that computing the weights as softmax of the advantages over the batch leads to good results in practice. During the high-level policy improvement, we found that clipping the value function between -100 and 0, which corresponds to the expected bounds given our choice of reward function and discount factor, stabilizes the training slightly.

KL divergence estimation. We use an exponentially moving average of the policy weights instead of the weights of the current policy to construct the prior policy π_{prior}: $\theta'_{k+1} = \tau\theta_k + (1-\tau)\theta'_k$ with the same smoothing coefficient τ as the one used for the Q function. We estimate the prior density using the following Monte-Carlo estimate:

$$\log \pi^{prior}(a|s,g) \approx \log\left[\frac{1}{I}\sum_i \pi_{\theta'}(a|s,s_g^i) + \epsilon\right], (s_g^i) \sim \pi_\psi^H(.|s,g), \tag{3.14}$$

where $\epsilon > 0$ is a small constant to avoid large negative values of the prior log-density. We use $I = 10$ samples to estimate $\pi^{prior}(a|s,g)$. We also use a Monte-Carlo approximation to estimate the KL-divergence term in Equation 3.11:

$$\begin{aligned} D_{\text{KL}}\left(\pi_\theta(.|s,g) \,||\, \pi^{prior}(.|s,g)\right) &= \mathbb{E}_{a\sim\pi(.|s,g)}[\log \pi_\theta(a|s,g) - \log \pi^{prior}(a|s,g)] \\ &\approx \frac{1}{N}\sum_n \left[\log \pi_\theta(a_n|s,g) - \log \pi^{prior}(a_n|s,g)\right] \\ \text{with } (a_n)_{n=1,..,N} &\sim \pi_\theta(.|s,g). \end{aligned} \tag{3.15}$$

Following SAC (Haarnoja et al., 2018b), we use $N = 1$, plug the estimate (3.14) and use the reparametrization trick to backpropagate the KL divergence term to the policy weights.

Experience relabelling. In all of our experiments we use Hindsight Experience Replay (Andrychowicz et al., 2017). We use the same relabelling strategy as Nair et al. (2018) and Nasiriany et al. (2019) and relabel the goals in our minibatches as follows:

— 20%: original goals from collected trajectories,
— 40%: randomly sampled states from the replay buffer,
— 40%: future states along the same collected trajectory.

Vision based environments On the vision-based robotic manipulation tasks, input images are passed through an image encoder shared between the policy, high-level policy and Q-function. Both states, goals and subgoals are encoded using the same encoder network. The encoder is updated during policy evaluation, where we only update the representation of the current state images whereas the representations of desired goal images, next state images and subgoal image candidates are kept fixed. We augment the observations with random translations by translating the 84×84 image within a 100×100 empty frame (Laskin et al., 2020; Kostrikov et al., 2020).

Hyperparameters Table 3.1 lists the hyperparameters used for the RIS. We use Adam optimizer and report results after one million interactions with the environment. For SAC, following Haarnoja et al. (2018c), we automatically tune the entropy of the policy to match the target entropy of $-\dim(\mathcal{A})$.

Table 3.1 – Hyper-parameters for RIS and SAC.

Hyper-parameter	Ant Navigation	Robotic Manipulation
Q hidden sizes	$[256, 256]$	$[256, 256]$
Policy hidden sizes	$[256, 256]$	$[256, 256]$
High-level policy hidden sizes	$[256, 256]$	$[256, 256]$
Hidden activation functions	ReLU	ReLU
Batch size	2048	256
Training batches per environment step	1	1
Replay buffer size	1×10^6	1×10^5
Discount factor γ	0.99	0.99
polyak for target networks τ	5×10^{-3}	5×10^{-3}
ϵ	1×10^{-16}	1×10^{-4}
Critic learning rate	1×10^{-3}	1×10^{-3}
Policy learning rates	1×10^{-3}	1×10^{-4}
High-level policy learning rate	1×10^{-4}	1×10^{-4}
α	0.1	0.1
λ	0.1	0.1

In the vision based environment, our image encoder is a serie of convolutional layers with kernel sizes $[3, 3, 3, 3]$, strides $[2, 2, 2, 1]$, channel sizes $[32, 32, 32, 32]$ and *ReLU* activation functions followed by a fully-connected layer with output dimension 16.

Table 3.2 – Environment specific hyper-parameters for LEAP

Hyperparameter	U maze	S maze	Π maze	ω maze	Robotic Manipulation
TDM policy horizon	50	50	75	100	25
Number of subgoals	11	11	11	11	3

For LEAP, we re-implemented Nasiriany et al. (2019) and train TDM (Pong et al., 2018) policies and Q networks with hidden layers of size $[400, 300]$ and $ReLU$ activation functions. In the ant navigation environments, we pretrain VAEs with mean squared reconstruction error loss and hidden layers of size $[64, 128, 64]$, $ReLU$ activation functions and representation size of 8 for the encoders and the decoders. In the vision based robotic manipulation environment, we pretrain VAEs with mean squared error reconstruction loss and convolutional layers with encoder kernel of sizes $[5, 5, 5]$, encoder strides of sizes $[3, 3, 3]$, encoder channels of sizes $[16, 16, 32]$, decoder kernel sizes of sizes $[5, 6, 6]$, decoder strides of sizes $[3, 3, 3]$, and decoder channels of sizes $[32, 32, 16]$, representation size of 16 and $ReLU$ activation functions. Table 3.2 reports the policy horizon used for each environment as well as the number of subgoals in the test configuration for the results in Figure 3.4. For the results presented in Figure 3.5, we adapted the number of subgoals according to the difficulty of each configuration.

3.4 Experiments

In this section we first introduce our experimental setup in Section 3.4.1. Next, we ablate various design choices of our approach in Section 3.4.3. We, then, compare RIS to prior work in goal-conditioned reinforcement learning in Section 3.4.2.

3.4.1 Experimental Setup

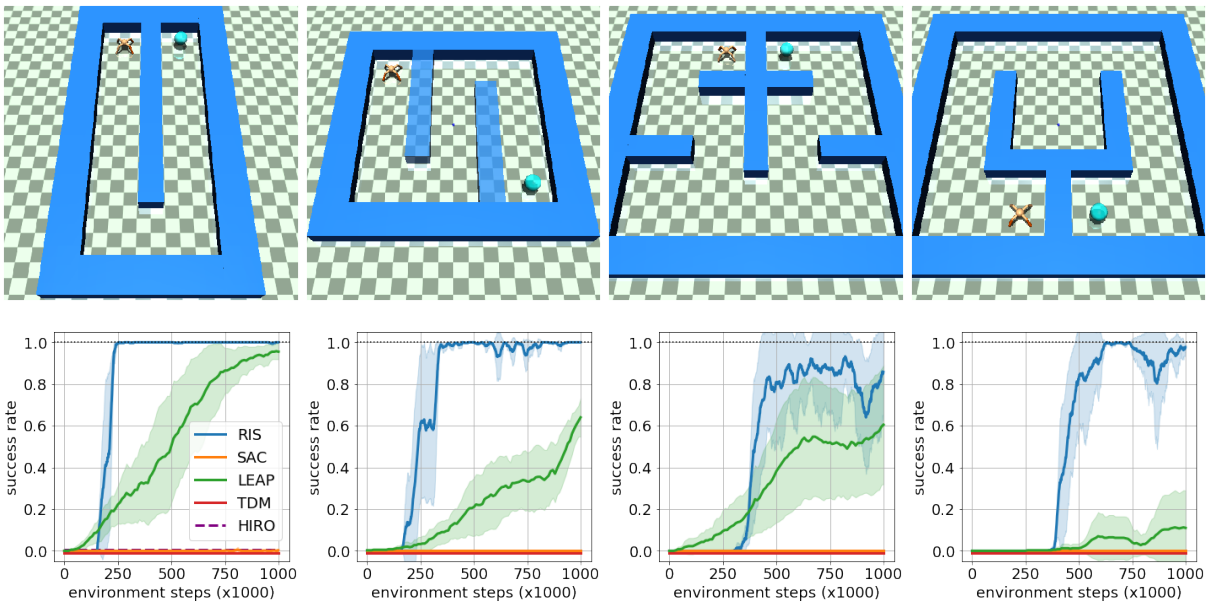
Ant navigation. We evaluate RIS on a set of ant navigation tasks of increasing difficulty, each of which requires temporally extended reasoning. In these environments, the agent observes the joint angles, joint velocity, and center of mass of a quadruped ant robot navigating in a maze. We consider four different mazes: a U-shaped maze, a S-shaped maze, a Π-shaped maze and a ω-shaped maze illustrated in Figure 3.4. The obstacles are unknown to the agent.

During training, initial states and goals are uniformly sampled and the agents are trained to reach any goal in the environment. We evaluate agents in the most extended temporal settings representing the most difficult configurations offered by the environment (see Figure 3.4). We assess the success rate achieved by these agents, where we define success as the ant being sufficiently close to the goal position measured by x-y Euclidean distance.

Vision-based robotic manipulation. We follow the experimental setup in Nasiriany et al. (2019) and also consider a vision-based robotic manipulation task where an agent controls a 2 DoF robotic arm from image input and must manipulate a puck positioned on

the table (Figure 3.6a). We define success as the arm and the puck being sufficiently close to their respective desired positions. During training, the initial arm and puck positions and their respective desired positions are uniformly sampled whereas, at test time, we evaluate agents on temporally extended configurations.

These tasks are challenging because they require temporally extended reasoning on top of complex motor control. Indeed, a greedy path towards the goal cannot solve these tasks. We train the agents for 1 million environment steps and average the results over 4 random seeds.



(a) U-shaped maze (b) S-shaped maze (c) Π-shaped maze (d) ω-shaped maze

Figure 3.4 – Comparison of RIS to several state-of-the-art methods (bottom row) on 4 different ant navigation tasks. We evaluate the success rate of the agent on the challenging configurations illustrates in the top row, where the ant is located at the initial state and the desired goal location is represented by a cyan sphere.

Alternative methods. We compare our approach, RIS, to off-policy reinforcement learning methods for goal-reaching tasks. We consider Soft Actor-Critic (SAC) (Haarnoja et al., 2018b) with HER, which trains a policy from scratch by maximizing the entropy regularized objective using the same sparse reward as ours. We also compare to Temporal Difference Models (TDM) (Pong et al., 2018) which trains horizon-aware policies operating under dense rewards in the form of distance to the goal. We chose to evaluate TDMs with a long policy horizon of 600 steps due to the complexity of considered tasks. Furthermore, we compare to Latent Embeddings for Abstracted Planning (LEAP) (Nasiriany et al., 2019), a competitive approach for these environments, which uses a sequence of subgoals

that a TDM policy must reach one after the other during inference.

We re-implemented SAC, TDM and LEAP and validated our implementations on the U-shaped ant maze and vision-based robotic manipulation environments.

On the U-shaped Ant maze environment, we additionally report results of HIRO (Nachum et al., 2018), a hierarchical reinforcement learning method with off-policy correction, after 1 million environment steps. The results were copied from Figure 12 in Nasiriany et al. (2019).

3.4.2 Comparison to the State of the Art

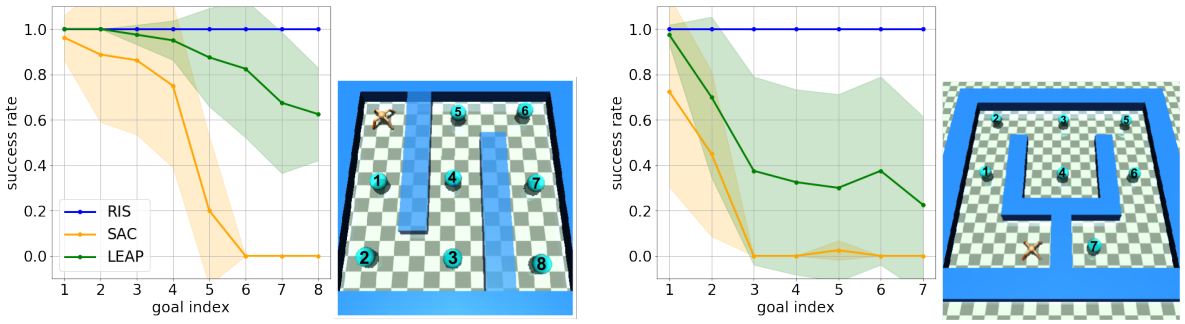
Ant navigation. Figure 3.4 compares RIS to the alternative methods introduced in section 3.4.1 for the four ant navigation environments. For all considered mazes RIS significantly outperforms prior methods in terms of sample efficiency, often requiring less than 500 thousand environment interactions to solve the mazes in their most challenging initial state and goal configurations.

LEAP makes progress on these navigation tasks, but requires significantly more environment interactions. This comparison shows the effectiveness of our approach, which uses subgoals to guide the policy rather than reaching them sequentially as done by LEAP. While SAC manages to learn goal-reaching behaviors, as we will see later in Figure 3.5, it fails to solve the environments in their most challenging configurations. The comparison to SAC highlights the benefits of using our informed prior policy compared to methods assuming a uniform action prior. On the U-shaped maze environment, HIRO similarly fails to solve the task within one million environment interactions. Furthermore, we observe that TDMs fails to learn due to the sparsity of the reward.

Figure 3.5 evaluates how SAC, LEAP and RIS perform for varying task horizons in the S-shaped and ω-shaped mazes. Starting from an initial state located at the edge of the mazes, we sample goals at locations which require an increasing number of environment steps to be reached. Figure 3.5 reports results for RIS, LEAP and SAC after having been trained for 1 million steps. While the performances of LEAP and SAC degrades as the planning horizon increases, RIS consistently solves configurations of increasing complexity.

These results demonstrate that RIS manages to solve complex navigation tasks despite relying on a flat policy at inference. In contrast, LEAP performs less well, despite a significantly more expensive planning of subgoals during inference.

Vision-based robotic manipulation. While the ant navigation experiments demonstrate the performances of RIS on environments with low-dimension state spaces, we also show how our method can be applied to vision-based robotic manipulation tasks. Our approach takes images of the current and desired configurations as input. Input images are passed through an image encoder, a convolutional neural network shared between



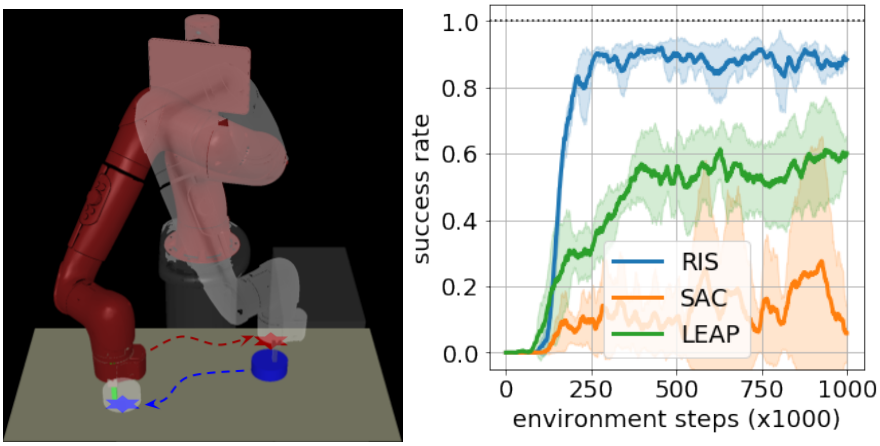
(a) S-shaped Maze (b) ω-shaped Maze

Figure 3.5 – Comparison of different methods for increasingly more difficult tasks on the S-shaped and ω-shaped ant maze. The goals with increasing complexity are numbered starting with 1, where 1 is the closest goal.

the policy, high-level policy and Q-function. The encoder is only updated when training the Q-function during policy evaluation and is fixed otherwise during policy improvement and high-level policy improvement. Instead of generating subgoals in the high-dimensional image space, the high-level policy therefore operates in the learned compact image representation of the encoder. Following recent works on reinforcement learning from images, we augment image observations with random translations (Kostrikov et al., 2020; Laskin et al., 2020). We found that using such data augmentation was important for training image-based RIS and SAC policies. Moreover, we found that using a lower learning rate for the policy was necessary to stabilize training. Additional implementation details on the image encoding are given in Appendix 3.3.5.

We compare our approach against LEAP and SAC in Figure 3.6b. RIS achieves a higher success rate than LEAP whereas SAC fails most of the time to solve the manipulation task consistently enough in the temporally extended configuration used for evaluation on the vision-based robotic manipulation task. Moreover, RIS and SAC only requires a single forward pass through their image encoder and actor network at each time step when interacting with the environment, whereas LEAP depends in addition upon an expensive planning of image subgoals.

Figure 3.7 visualizes the imagined subgoals of the high-level policy. Once the RIS agent is fully trained, we separately train a decoder to reconstruct image observations from their learned representations. Given observations of the current state and the desired goal, we then predict the representation of an imagined subgoal with the high-level policy and generate the corresponding image using the decoder. Figure 3.7 shows that subgoals predicted by the high-level policy are natural intermediate states halfway to the desired goal on this manipulation task. For example, for the test configuration (Figure 3.7 top), the high-level policy prediction corresponds to a configuration where the arm has reached



(a) Illustration of the robotic manipulation task

(b) Comparison to prior works

Figure 3.6 – Robotic manipulation environment: (a) illustration of the task; (b) results of our method compared to LEAP and SAC.

the right side of the puck and is pushing it towards its desired position.

3.4.3 Ablative Analysis

We next discuss and evaluate several design choices of our method. We use the Ant U-maze navigation task and the robotic manipulation environment for ablations, and evaluate the quality and the impact of subgoals in Sections 3.4.3.1-3.4.3.2. The impact of data augmentation and the choice of learning rates are evaluated in Sections 3.4.3.3 and 3.4.3.4 respectively. Finally we evaluate the learning strategy for the prior policy on the Ant navigation tasks in Section 3.4.3.5.

3.4.3.1 Imagined subgoals

To evaluate the quality of the subgoals predicted by our high-level policy, we introduce an oracle subgoal sampling procedure. We plan an oracle trajectory, which corresponds to a point-mass agent navigating in this maze and does not necessarily correspond to the optimal trajectory of an ant. Oracle subgoals correspond to the midpoint of this trajectory between state and goal in x-y location (Figure 3.8 left).

Figure 3.8 (left) shows the subgoal distribution predicted by a fully-trained high-level policy for a state and goal pair located at opposite sides of the U-shaped maze. We can observe that its probability mass is close to the oracle subgoal.

To quantitatively evaluate the quality of imagined subgoals, we measure the x-y Euclidean distance between oracle subgoals and subgoals sampled from the high-level policy

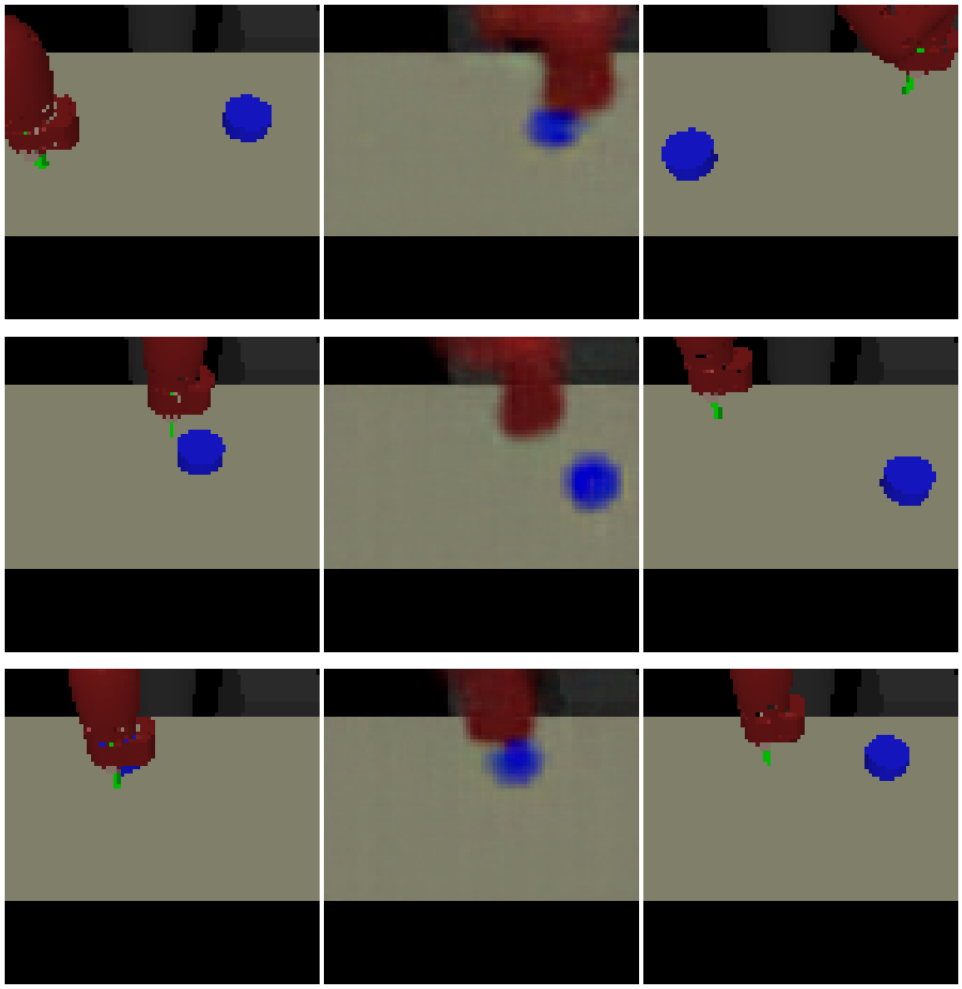


Figure 3.7 – Image reconstruction of an imagined subgoal (middle) given the current state (left) and the desired goal (right) on a temporally extended configuration used for evaluation (top) and a random configuration (bottom).

throughout training for a set of fixed state and goal tuples randomly sampled in the environment. Figure 3.8 (right) shows that RIS successfully learns to find subgoals that are coherent with the oracle trajectory during training, despite not having prior knowledge about the environment.

We also assess the importance of using the implicit regularization scheme presented in Section 3.3.2 which discourages high-level policy predictions to lie outside of the distribution of valid states. We compare our approach against naively optimizing subgoals without regularization (i.e. directly optimize (3.4)). Figure 3.8 (right) shows that without implicit regularization, the predicted subgoals significantly diverge from oracle subgoals in x-y location during training. As result, imagined subgoals are not able to properly guide policy learning to solve the task, see Figure 3.9.

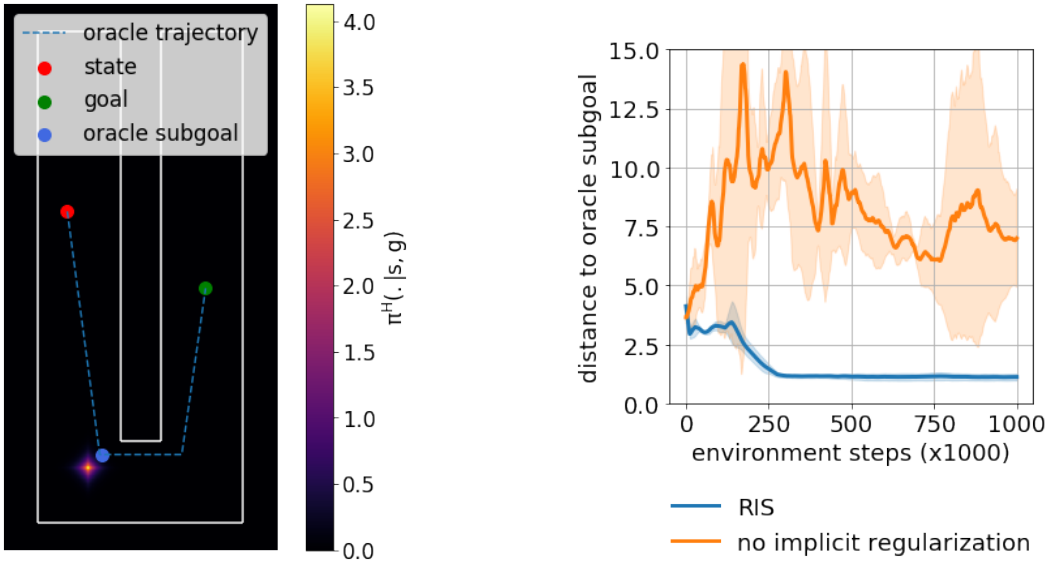


Figure 3.8 – (Left) Heatmap of the subgoal distribution obtained with our high-level policy and the oracle subgoal for a given state and goal for the ant U-maze environment. (Right) Distance between oracle subgoals and subgoals predicted by the high-level policy for RIS and RIS without implicit regularization. The dimensions of the space are 7.5×18 units and the ant has a radius of roughly 0.75 units.

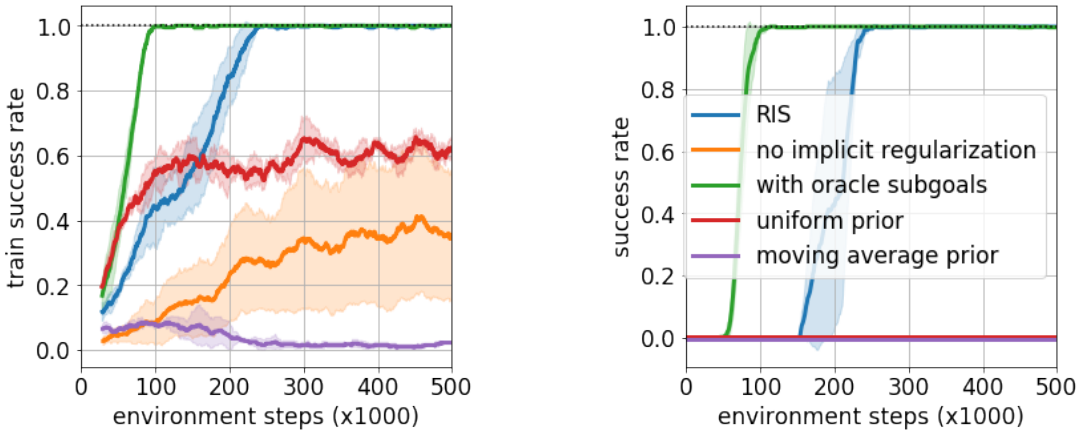


Figure 3.9 – Ablation of our method on the Ant U-Maze environment: simple priors that do not incorporate subgoals (*uniform prior, moving average prior*); ignoring the effect of out-of-distribution subgoal predictions (*no implicit regularization*); and using oracle subgoals (*with oracle subgoals*). Left: success rate on all configurations throughout training. Right: success rate on the test configurations.

3.4.3.2 Prior policy with imagined subgoals

We evaluate the importance of incorporating imagined subgoals into the prior policy by comparing to a number of variants. To disentangle the effects of imagined subgoals from the actor-critic architecture used by RIS, we first replace our prior policy with simpler choices of prior distributions that do not incorporate any subgoals: (i) a uniform prior over the actions $\pi_k^{prior} = \mathcal{U}(\mathcal{A})$, which is equivalent to SAC without entropy regularization

during policy evaluation, and (ii) a parametric prior policy that uses an exponential moving average of the online policy weights $\pi_k^{prior} = \pi_{\theta'_k}$. Figure 3.9 shows that, while they can learn goal-reaching behaviors on many configurations encountered during training (left), neither of these variants are able to solve the Ant U-maze environment in its most difficult setting (right). We also observe that the agent with a moving average action prior fails to learn (left). This highlights the benefits of incorporating subgoals into policy learning.

Finally, we propose to incorporate the oracle subgoals into our prior policy. We replace the subgoal distributions predicted by our high-level policy with Laplace distributions centered around oracle subgoals. Results in Figure 3.9 show that RIS with oracle subgoals learns to solve the U-shaped ant maze environment faster than using a high-level policy simultaneously trained by the agent. This experiment highlights the efficiency of our approach to guide policy learning with appropriate subgoals: if we have access to proper subgoals right from the beginning of the training, our approach could leverage them to learn even faster. However, such subgoals are generally not readily available without prior knowledge about the environment. Thus, we introduce a high-level policy training procedure which determines appropriate subgoals without any supervision.

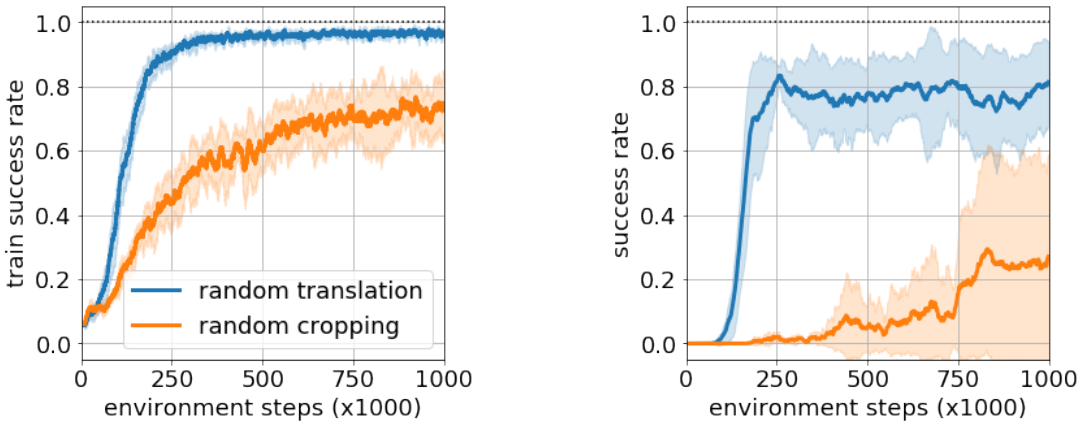
3.4.3.3 Data augmentation

Data augmentation is a standard practice to cope with image variability. We study the effect of image augmentation when applying RIS to the vision-based robotic manipulation environment. We consider two types of image augmentations that have proven effective in Kostrikov et al. (2020) and Laskin et al. (2020):

1. Random cropping, where we first resize the observation in a larger frame of size 100×100 then extract a random patch of the original 84×84 size
2. Random translation, where we first pad each side of the observation with 8 pixels then extract a random patch of the original 84×84 size

As we see in Figure 3.10, augmenting the image observations with random cropping performs worse than with random image translations. We hypothesis that changing the scale of the observation when performing random cropping, as opposed to random translations, hurts the estimation of the relative positions of the arm and the puck, which is detrimental for the success of the policy on this manipulation task. In contrast, random image translation preserves the relative positions of objects and results in significant improvements.

Next, we ablate the strength of the random translation applied to the 84×84 image observations. We pad each image side by either 0, 4, 8 or 16 pixels and then randomly crop the resulting view back to 84×84 size.



(a) Train success rate

(b) Test success rate

Figure 3.10 – We compare image augmentations in terms of random cropping and random translation when applying RIS to the vision-based robotic manipulation environment.

Figure 3.11 illustrates that such image augmentation is essential for training successful policies and confirms previous findings in Kostrikov et al. (2020) and Laskin et al. (2020). For training configurations with uniformly sampled puck and arm positions on the table, different levels of augmentation result in similar success rates, while stronger augmentations imply slower training, see Figure 3.11(a).

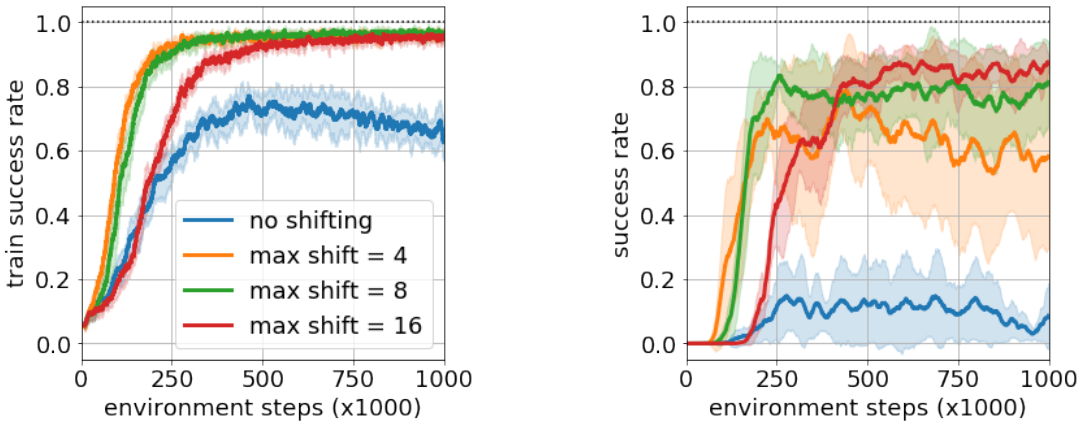
Meanwhile, for the more challenging test configurations, the agent achieves higher success rates when using stronger image augmentations, see Figure 3.11(b). This underlines a trade-off between sample efficient learning of low-level control skills to solve the training tasks, and learning more robust representations for the temporally extended tasks, where imagined subgoals are necessary to obtain good performances. Based on these results, we use random translations with 8 pixels padding for experiments with the vision-based manipulation task presented in Sections 3.4.2 and 3.4.3.4.

3.4.3.4 Learning rates

We investigate the influence of learning rates when optimizing the policy and the critic for the vision-based robotic manipulation task. Figures 3.12,3.13 present results for a set of learning rates $\{1e-3, 1e-4, 3e-5\}$. As can be observed, it is preferable to use lower learning rates for policy learning compared to learning rates for the critic. Too low learning rates, however, result in slower training.

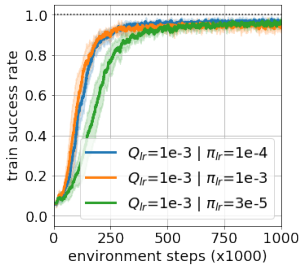
3.4.3.5 Exponential moving average policy versus Boltzmann policy

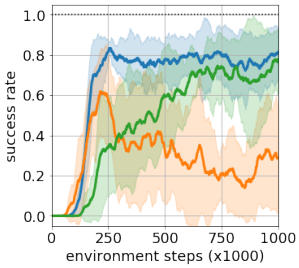
We compare the two approaches for incorporating imagined subgoals into policy learning described in Section 3.3.3. Figure 3.14 shows experiments



(a) Train success rate (b) Test success rate

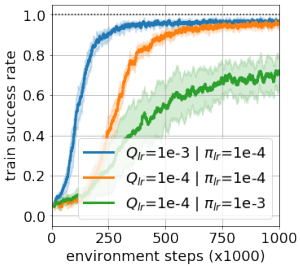
Figure 3.11 – We ablate the strength of image translation applied during training. We plot success rates achieved on the training configurations (left) and on the evaluation configurations (right).

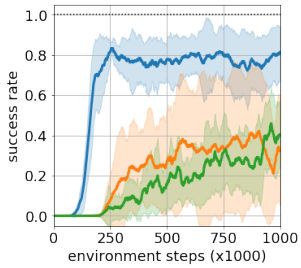




(a) Train success rate (b) Test success rate

Figure 3.12 – Policy learning rate





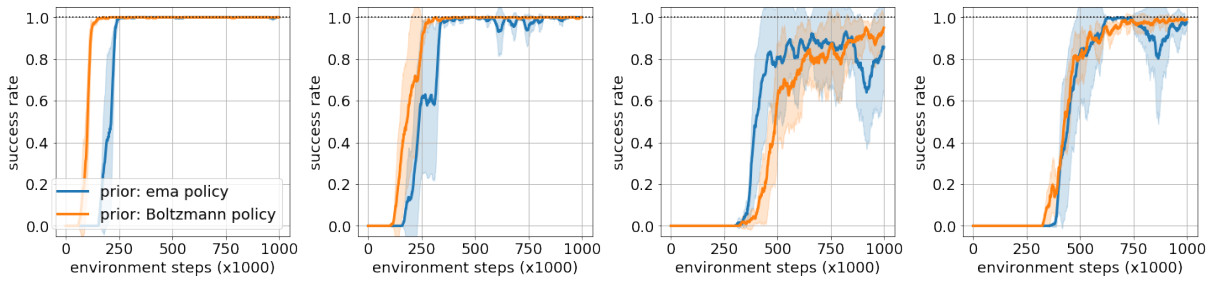
(a) Train success rate (b) Test success rate

Figure 3.13 – Critic learning rate

1. using an Exponential Moving Average (EMA) of the online policy weights as the prior policy, where the policy update is defined by (3.11),
2. defining the prior policy in terms of the Boltzmann policy associated with the learned Q function, as defined in (3.13).

Although the EMA policy, being more representative of the online policy distribution, might provide stronger guidance, its narrower distribution compared to the Boltzmann policy may imply larger KL-divergences and instabilities.

While both methods manage to solve all the mazes in their most challenging configurations, using the Boltzmann policy achieves higher sample efficiency compared to the EMA policy on the U and S shaped mazes.



(a) U-shaped maze (b) S-shaped maze (c) Π-shaped maze (d) ω-shaped maze

Figure 3.14 – Comparison of the exponential moving average policy (*EMA policy*) with the Q-function (*Boltzmann policy*) for constructing the prior policy in RIS.

3.5 Conclusion

We introduced RIS, a goal-conditioned reinforcement learning method that imagines possible subgoals in a self-supervised fashion and uses them to facilitate training. We propose to use the value function of the goal-reaching policy to train a high-level policy operating in the state space. We then use imagined subgoals to define a prior policy and incorporate this prior into policy learning. Experimental results on challenging simulated navigation and vision-based manipulation environments show that our proposed method greatly accelerates learning of temporally extended tasks and outperforms competing approaches.

While our approach makes use of subgoals to facilitate policy search, future work could explore how to use them to obtain better Q-value estimates. Future work could also improve exploration by using imagined subgoals to encourage the policy to visit all potential states.

Chapter 4

Learning Video-Conditioned Policies for Unseen Manipulation Tasks

In the previous chapter, we introduced a method to enhance goal-conditioned reinforcement learning for long-horizon tasks. Nevertheless, a task definition solely based on goal configurations has its limitations. In this chapter, we introduce an alternative approach that leverages human videos to convey the intended task to the robot.

Indeed, the ability to specify robot commands by a non-expert user is critical for building generalist agents capable of solving a large variety of tasks. One convenient way to specify the intended robot goal is by a video of a person demonstrating the target task. While prior work typically aims to imitate human demonstrations performed in robot environments, here we focus on a more realistic and challenging setup with demonstrations recorded in natural and diverse human environments.

We propose *Video-conditioned Policy learning (ViP)*, a data-driven approach that maps human demonstrations of previously unseen tasks to robot manipulation skills. To this end, we learn our policy to generate appropriate actions given current scene observations and a video of the target task. To encourage generalization to new tasks, we avoid particular tasks during training and learn our policy from unlabelled robot trajectories and corresponding robot videos. Both robot and human videos in our framework are represented by video embeddings pre-trained for human action recognition. At test time we first translate human videos to robot videos in the common video embedding space, and then use resulting embeddings to condition our policies.

Notably, our approach enables robot control by human demonstrations in a *zero-shot manner*, i.e., without using robot trajectories paired with human instructions during training. We validate our approach on a set of challenging multi-task robot manipulation environments and outperform state of the art. Our method also demonstrates excellent

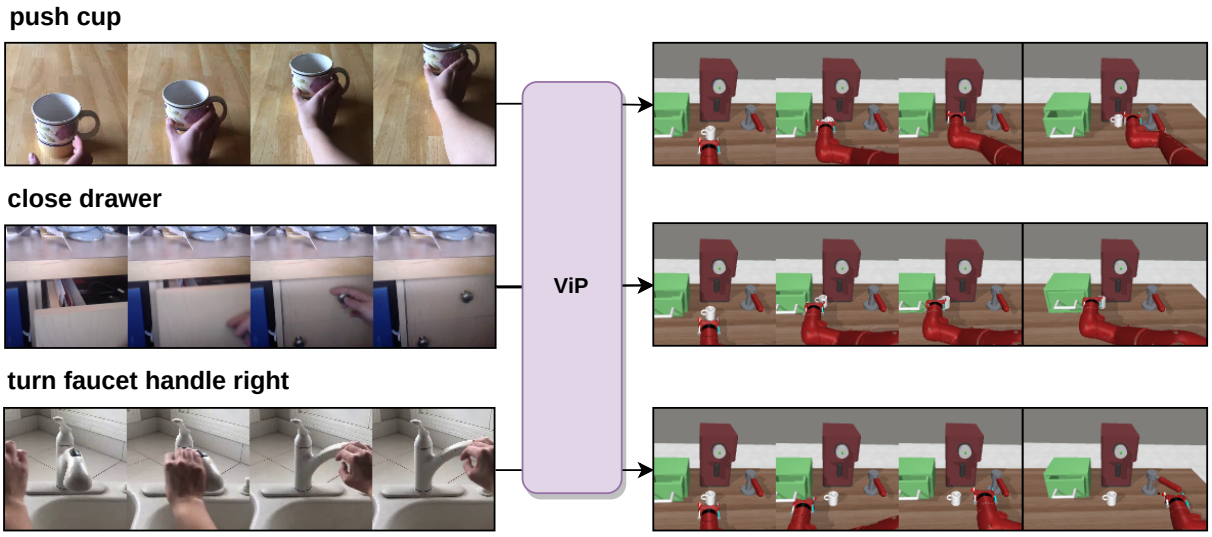


Figure 4.1 – Given a human video instruction in a non-robotic scene, our video-conditioned policy ViP controls the robot to perform a similar task zero-shot, i.e. the agent never observes robot data paired with human instructions during training and figures out which manipulation skill it is expected to perform in its environment at test time. We illustrate three examples of different tasks demonstrated by people and the corresponding roll-outs generated by our method for the TableTop robotics environment.

performance in a new challenging zero-shot setup where no paired data is used during training.

4.1 Introduction

Significant progress has been made in recent years towards learning a generalist robot agent capable of accomplishing a wide array of skills across many environments (Ahn et al., 2022; Jang et al., 2022; Lu et al., 2022; Chebotar et al., 2021). Central to this challenge is the ability to effectively specify tasks and rewards to the robot system in a user-friendly manner. In reinforcement learning, a task is commonly defined through a reward function (Sutton and Barto, 2018). However, designing good reward functions for each task is often challenging and restricting policy learning to a fixed set of tasks hinders generalization to new tasks. Goal-conditioned imitation and reinforcement learning (Ghosh et al., 2019; Lynch et al., 2020; Eysenbach et al., 2020b; Nair et al., 2018; Gupta et al., 2019b; Pong et al., 2018) can learn agents capable of performing a wide diversity of tasks with less supervision. But providing the right goal to define the task requires expert operators to come up with a suitable robot observation of the desired configuration. Other works have shown that we can learn to command generalist robots through language instructions (Jang et al., 2022; Lynch and Sermanet, 2020; Nair et al.,

2022a; Mees et al., 2022a) and human videos (Jang et al., 2022; Chen et al., 2021a), which are easy to provide for non-expert operators and can generalize to unseen inputs, including behaviors beyond goal-reaching skills. Furthermore, being able to specify robot skills through language or video commands unlocks solving more complex long-horizon tasks by chaining human instructions (Ahn et al., 2022; Mees et al., 2022b). Nonetheless, these methods rely on annotated demonstrations for a large set of robot skills, which is often tedious to provide, especially since task annotation must be repeated for each new robot environment.

In this work, we propose ***Vi****deo-conditioned* ***P****olicy learning* (ViP), a method that learns to perform manipulation skills given a human video of the desired task in vision-based multi-task robotic environments (Figure 4.1). We demonstrate that, due to the similarity between robot manipulation and videos of humans performing manipulation skills, we can leverage existing large datasets of annotated human videos, such as the Something-Something-v2 dataset (Goyal et al., 2017) (SSv2), to learn to map human videos to robot behaviors *in a zero-shot manner*, without training on paired data between human instructions and robot demonstrations. For instance, the robot may interact in an environment that includes a drawer, yet has received no supervision on what it is expected to do when commanded with a video instruction of a human closing a possibly different drawer. We do so by learning a video encoder using Supervised Contrastive Learning (Khosla et al., 2020), where embeddings of videos of the same task are closer together in cosine distance, and show that video models trained on such large datasets of annotated human videos, which can be easily collected from the internet (Goyal et al., 2017; Miech et al., 2019; Grauman et al., 2022; Damen et al., 2018), generalize to the robot video domain.

In addition, we show that video encoders trained for human video action recognition readily provide relevant task embeddings for multi-task policy learning. Following recent trends in data-driven robotics, our approach learns from large collections of offline robot experience that can be collected in different ways, e.g. by expert demonstrations given by motion planners, teleoperated play data, random data generation processes. Given an offline dataset of robot demonstrations, we learn a video conditioned policy to regress the action conditioned on both the robot state and a video embedding of the full trajectory the state-action pair belongs to. We also keep each embedding in a library of robot embeddings. At inference, we perform nearest neighbors regression on this library using the cosine distance to the embedding of the human instruction to generate an appropriate robot embedding that is both (a) relevant to the instruction and (b) is executable by the policy. We can then execute the human video instruction by decoding the selected embedding into a robot trajectory using the learned policy.

Overall, our approach demonstrates that large collections of human videos can enable less supervision in data-driven robotics. Our contributions can be summarizes as follows:

— we designed a method to train a policy conditioned on robot video embeddings given by a video encoder pretrained on a large dataset of annotated human videos;

— our method can map human video instructions to robot manipulation skills without supervision from paired data to bridge the gap between the human and robot domain;

— our approach outperforms prior works on a set of multi-task robotic environments. [1]

4.2 Related Work

Different methods have been explored in recent years towards robot learning with human videos. Many prior works hence consider the problem of learning to follow human demonstrations using different techniques such as pose or keypoints estimation (Peng et al., 2018b; Xiong et al., 2021; Das et al., 2020), image translation and inpainting (Liu et al., 2018; Sharma et al., 2019; Smith et al., 2019; Bahl et al., 2022), learning object centric representations (Pirk et al., 2019; Sermanet et al., 2018), simulators (Petrik et al., 2020; Bonardi et al., 2020) and meta learning (Yu et al., 2018). Contrary to these prior works that often consider closely aligned human videos and robot environments e.g. humans demonstrating the task in the same lab environment as the robot, we assume that human video instructions are collected "in-the-wild" and therefore there exists a large domain gap between human videos and robot workspace.

Recent works attempt to perform model pretraining from large-scale human videos in order to get good image representations for robotic control (Nair et al., 2022b; Xiao et al., 2022). Other works have shown that we can infer states and actions from diverse videos and use it for reinforcement learning (Edwards and Isbell, 2019; Schmeckpeper et al., 2020a,b; Seo et al., 2022). In our work, we show that encoders trained on large datasets of videos for human action recognition provide good task embeddings for robotic manipulation.

Prior works have also considered leveraging large datasets of human videos to learn reward functions for robotics manipulations (Shao et al., 2021; Chen et al., 2021a). The most relevant work to ours is Domain-agnostic Video Discriminator (DVD) (Chen et al., 2021a) which also tackles the problem of commanding robots using in-the-wild human videos. DVD learns a video similarity by training a discriminator network to classify whether two videos are performing the same task on both annotated robot demonstrations

1. Project website: https://www.di.ens.fr/willow/research/vip/.

and a subset of SSv2 videos. The similarity score is then used as a reward for planning with an action-conditioned video generation method (Babaeizadeh et al., 2017) trained on randomly collected robot experience. In contrast, our approach does not require any annotated robot demonstration and can accommodate both randomly collected robot experience and expert demonstrations. Moreover, Chen et al. (2021a) plans several sub-trajectories as high-dimensional synthetic videos per episode whereas our approach plans a full trajectory in the embedding space of robot videos.

Contrastive learning on large-scale datasets has led to significant progresses in a range of computer vision tasks (Chen et al., 2020; He et al., 2020). Contrastive learning has also been used to learn language-conditioned policies by training on large scale datasets of images (Radford et al., 2021; Shridhar et al., 2022) and videos (Fan et al., 2022). In this work, we leverage Supervised Contrastive Learning (Khosla et al., 2020) for human action recognition on the SSv2 dataset to learn a mapping between human video instructions and robotic manipulation.

Our work falls under outcome-conditioned action regression (Lynch et al., 2020; Emmons et al., 2021; Furuta et al., 2021). Such works cast reinforcement learning as learning policies conditioned on trajectory information such as future returns (Kumar et al., 2019b; Schmidhuber, 2019; Srivastava et al., 2019), many previous timesteps (Chen et al., 2021b; Janner et al., 2021) or a desired goal-configuration (Ding et al., 2019; Ghosh et al., 2019; Lynch et al., 2020; Emmons et al., 2021) to learn multi-task policies. In contrast, our policy is conditioned on a video embedding of the full robot trajectory. (Jang et al., 2022) also showed that we can learn a video-conditioned controllers by conditioning the policy on both a video of the full robot trajectory and paired sets of human videos appropriately collected for each task. Other works (Lynch and Sermanet, 2020; Nair et al., 2022a) show that pairing a small subset of robot data with language instructions enables generalizable language-conditioned task execution. Our method does not require any robot data paired with human instructions and, hence, can learn generic policies from unlabelled robot datasets.

4.3 Method

In Section 4.3.1 we first present an overview of our approach that enables robots to mimic new tasks demonstrated by people in natural human environments. We then detail how we learn a generic video-conditioned policy from randomly-generated demonstrations in Section 4.3.2. We further describe how we condition our policy on human videos in Section 4.3.3. Finally, we describe how we learn a similarity function for matching human videos to robot roll-outs without using paired human-robot data in Section 4.3.4.

4.3.1 Method overview

We aim to perform a robot manipulation task conditioned on a human video in a vision-based multi-task environment. At test time our system receives an input video of a previously unseen task performed by a person, such as pushing a mug or closing a drawer, and controls a robot arm with the intent of performing a similar task in the robot environment.

More formally, we consider a set of Markov Decision Processes (MDP) $\mathcal{MDP}_i = (\mathcal{S}, \mathcal{A}, \mathcal{R}_i, p)_i$ sharing the same observation space $\mathcal{S}$, action space $\mathcal{A}$ and dynamics p but with different reward functions $\mathcal{R}_i$ corresponding to different tasks we would like to solve. We do not assume that the reward functions $\mathcal{R}_i$ are observed. Instead, $\mathcal{R}_i$ must be inferred through a human video of the task $x^h \in \mathcal{X}$. Our goal is to learn a video-conditioned controller $\pi(.|s, x^h)$ that predicts actions $a \in \mathcal{A}$ given the current states $s \in \mathcal{S}$ and human videos $x^h \in \mathcal{X}$ to maximize the reward functions associated with the input human video.

There may exist a large domain gap between videos of humans and robots performing similar tasks. To bridge this gap, we leverage the large-scale video dataset Something-Something-v2 (SSv2) (Goyal et al., 2017) with labeled human actions $D^h = \{x_i^h, y_i^h\}_i$ where labels y_i^h for videos x_i^h correspond to different manipulation actions such as Opening Something, Moving Something Away from the Camera, etc. Following DVD (Chen et al., 2021a), we train a similarity function $d(.,.)$ that assigns high values to pairs of videos representing the same task and low values to video pairs of different tasks. Unlike DVD, however, we learn such similarity without any annotated robot videos. Given a human video, we use the learned similarity as a reward $R(.) = d(., x^h)$ for our controller.

To learn the policy, we assume access to a dataset of unlabelled robot demonstrations $D^r = \{x_i^r, (s_i^t)_t, (a_i^t)_t\}_i$, where $(s_i^t)_t$ is a sequence of robot observations, $(a_i^t)_t$ is the corresponding sequence of executed robot actions and x_i^r is a video of the demonstration. For instance, if the observation space $\mathcal{S}$ only contains images (without e.g. proprioceptive information), the videos can simply correspond to the whole sequence of states in the trajectory $x_i^r = (s_i^t)_t$. This offline dataset can be collected in many different ways: by expert demonstrators, rollouts of other policies, through teleoperation or by random data generation. Importantly, we do not assume access to any further information about these demonstrations.

Figure 4.2 presents an overview of our approach: we leverage a video encoder f_θ trained for human action recognition on SSv2 and a similarity metric between videos. During training, we learn a behavior cloning policy conditioned on robot embeddings given by the video encoder while storing all embeddings of the robot training dataset in a library. At inference, we first use this library and the similarity metric to predict a robot embedding relevant to the human video instruction, then rollout the policy in the

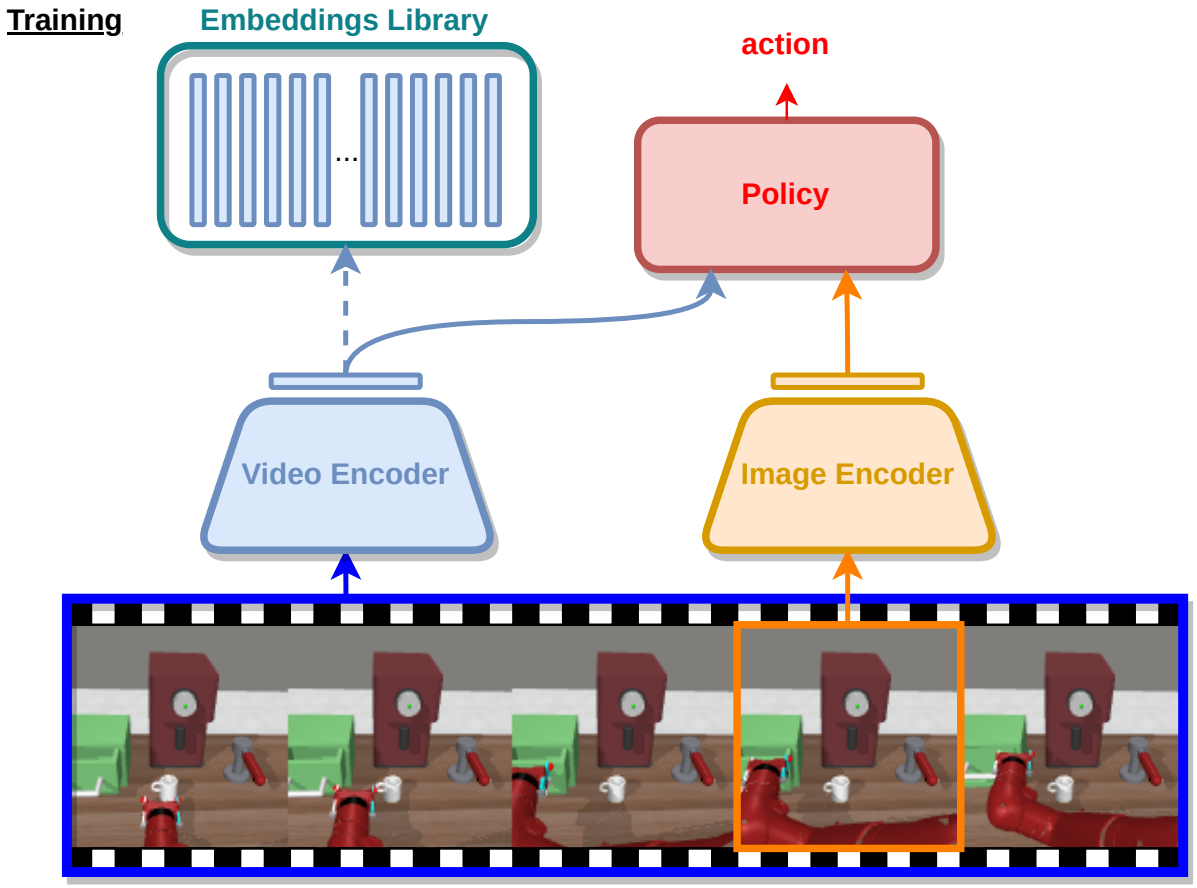


Figure 4.2 – During training, we learn a manipulation policy conditioned on robot video embeddings of the full robot trajectories from the robot dataset. At the same time, the robot video embedding of each trajectory in the robot dataset is added to an embeddings library.

environment.

4.3.2 Video-conditioned policy learning

We now describe how to encode a large array of behaviors into a single policy. A key to our approach is the use of the embedding space of a video encoder pretrained on human videos to condition our policy. This video embedding can be seen as a task embedding for our generic multi-task policy. We will explain more in detail how we can obtain meaningful video embeddings for control in Section 4.3.4.

During training, we learn a policy π_ϕ to regress an action given the current state and a video embedding of a full robot trajectory. Video embeddings act as context defining the global task. The policy is trained with behavior-cloning by minimizing the loss:

$$\mathcal{L}_\pi(\phi) = -\mathbb{E}_{s,a,x^r \sim D^r} \log \pi_\phi(a|s, f_\theta(x^r)). \tag{4.1}$$

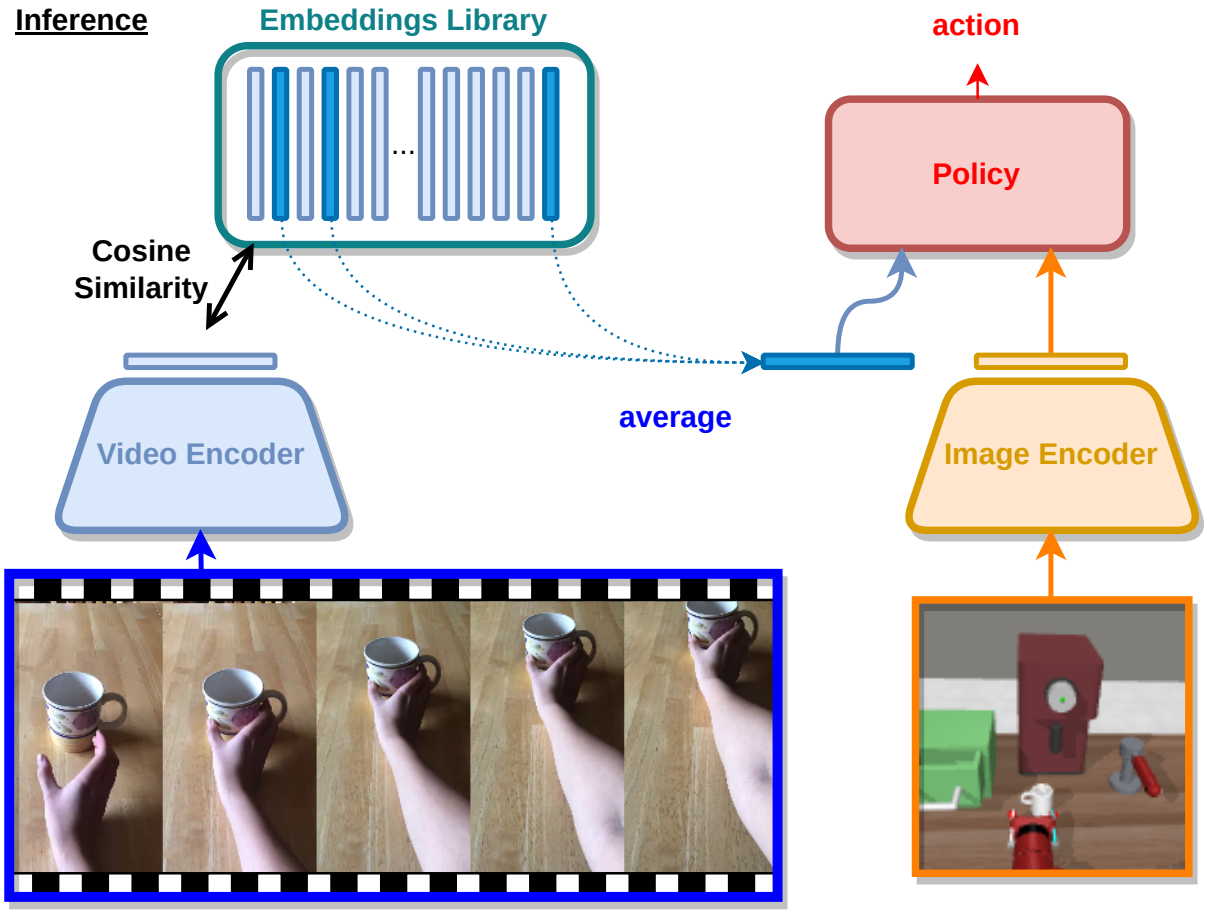


Figure 4.3 – At inference, we encode the human video instruction into a human video embedding. We then average the robot embeddings from the library that have highest cosine similarity to the human embedding into a selected robot embedding. Finally, we execute the policy conditioned on this selected embedding.

At test time, this policy can be commanded to reproduce a robot video input x^r by first encoding the video into an embedding e^r and then executing the actions $a \sim \pi(.|s, e^r)$ predicted by the policy at each step s visited during the roll-out. As we will see from experiments, however, using human videos to directly condition the policy results in poor performance due to the large domain gap between robot and human videos. We therefore propose to first translate human videos to robot video embeddings as described in the next section.

4.3.3 Inference with human instructions

At inference, our first step is to translate the human video instruction x^h into the robot video embedding e^h that both (1) corresponds to the target task and (2) is in distribution for the robot policy. While many methods can instantiate this, we choose to

simply use nearest neighbors regression in the robot embedding space. We first encode the videos contained in the robot dataset D^r into a library of robot embeddings $D^e = (e_i^r = f_\theta(x_i^r))_i$. We also encode the video instruction into a human embedding e^h. We then perform k nearest neighbors regression using the distance function d, by first computing all the distances between the human embedding and each embeddings in the library $d_i = d(e^h, e_i^r)$, then averaging the top k embeddings of the library $e^r = \frac{1}{k}\sum_{i\in 1}^{k} e_i^r$. Finally, we perform policy rollout conditioned on this embedding. As result, although the policy was trained with behavior cloning, this approach allows us to maximize the similarity of the robot trajectory to the video prompt at inference.

4.3.4 Learning a task similarity from human videos

Many choices of distance metric d between videos can be used to match a human video to an appropriate robot embedding. In this work, we consider adapting Supervised Contrastive Learning (Khosla et al., 2020) to our video action recognition task on the Something-Something-v2 dataset. We learn our video encoder $f_\theta(.) = f_\theta^p(f^b(.))$ composed of a backbone $f^b(.)$ which maps videos to representation vectors and a projection network $f_\theta^p(.)$ which maps representation vectors to embedding vectors e, such that embeddings from the same class are pulled closer together in cosine distance than embeddings from different classes. As a result, the cosine distance characterizes the similarity between two videos.

In contrastive representation learning, the backbone and projection net are typically trained end-to-end from scratch and, after training, the projection net is discarded and a classifier head is learned on top of the representations. Instead, we start from available pretrained backbones for SSv2 classification and simply train the projection net using the Supervised Contrastive Learning loss. Given a batch of N video/label pairs sampled from the SSv2 dataset $\{x_k^h, y_k^h\}_{k\in[1,..,N]} \sim D^h$, we build a multiview batch consisting of $2N$ pairs, $\{\tilde{x}_l^h, \tilde{y}_l\}_{l\in[1,..,2N]}$, where $\tilde{x}_{2k-1}^h$ and $\tilde{x}_{2k}^h$ are two random augmentations of video x_k^h and $\tilde{y}_{2k-1}^h = \tilde{y}_{2k}^h = y_k^h$. We train f_θ to minimize the Supervised Contrastive Loss:

$$\mathcal{L}_{SupCon}(\theta) = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(\langle f_\theta(\tilde{x}_i^h), f_\theta(\tilde{x}_p^h)\rangle/\tau\right)}{\sum_{a\in A(i)} \exp\left(\langle f_\theta(\tilde{x}_i^h), f_\theta(\tilde{x}_a^h)\rangle/\tau\right)} \tag{4.2}$$

where $\langle .,. \rangle$ is the cosine similarity, $I = [1,..,2N]$, $A(i) = I\backslash\{i\}$, $P(i) = \{p \in A(i) : \tilde{y}_p^h = \tilde{y}_i^h\}$ and τ is a hyperparameter. After training, we use the distance $d(x^h, x^r) = \langle f_\theta(x^h), f_\theta(x^r)\rangle$ between videos x^h and x^r as measure of similarity that focuses on the semantic aspects of the video. As we show in the experimental section, despite being trained only on human videos, this similarity metric generalizes to robot manipulation

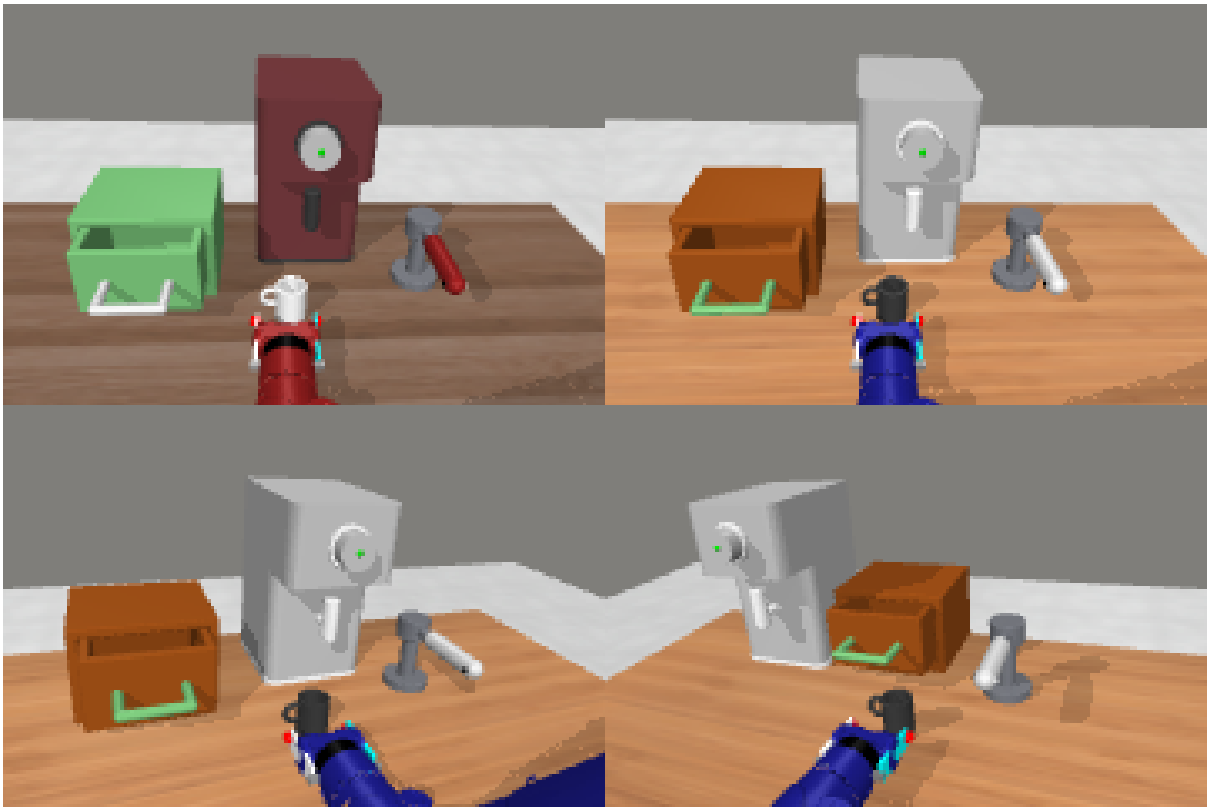


Figure 4.4 – Illustrations of the TableTop environments used in our experiments. Top row: env1 and env2. Bottom row: env3 and env4.

tasks. In our experiments, we use the same video backbone as Shao et al. (2021) and Chen et al. (2021a).

Alternative options for d include the DVD similarity network (Chen et al., 2021a), which learns to classify whether two videos correspond to the same task and uses the classification score between human and robot videos as a distance metric d on top of the video encoder f^b. While our approach implicitly corresponds to the same classification objective, supervised contrastive learning readily embeds our encoder with a similarity metric.

4.4 Experiments

This section presents experiments validating our proposed approach and its training procedure. We evaluate our method on several tasks in two challenging environments and compare results to the state-of-the-art method (Chen et al., 2021a). In particular, we present ablations of our method and show its advantage in zero-shot settings, i.e. for tasks that have not been observed during training.

4.4.1 Experimental Setup

For our experiments, we consider the TableTop environments introduced in Chen et al. (2021a). The agent controls a robot arm in four variations of a simulated environment containing a drawer, a cup in front of a coffee machine and a faucet handle. The four environments differ by object positions, camera locations and colors. Figure 4.4 illustrates the four variations of the TableTop environment. Robot experience for policy training is collected by controlling the robot end effector to go through three keypoints randomly sampled in the environment, as illustrated in figure 4.5. As a result, certain demonstrations may exhibit semantically useful behaviors, while the majority of them do not.

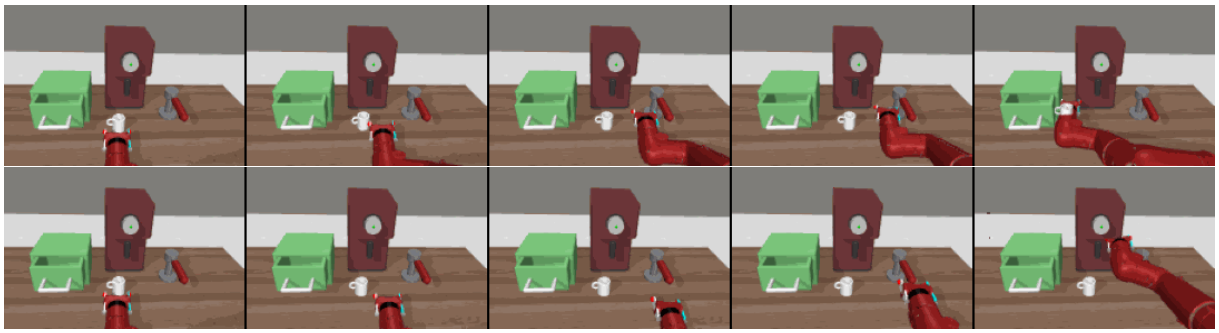
We follow the experimental setup of Chen et al. (2021a) and consider three tasks: close the drawer, push the cup and turn the faucet to the right. The image encoder representing the current state of the environment is learned end-to-end together with the policy. These environments challenge the ability of our approach to generate meaningful manipulation skills from random robot trajectories.

Furthermore, we consider the Kitchen environment initially introduced in Gupta et al. (2019b). The agent controls a robot arm with a clamp in a kitchen environment containing diverse objects: a microwave, three doors, a kettle and light and knob switches. Figure 4.6 illustrates the two variations of the Kitchen environment, which differ by the position of the camera in the scene. We consider 3 opening tasks: open the microwave, open the left door and open the sliding door. Robot demonstrations accomplishing these tasks are available from Nair et al. (2022b). However, these demonstrations are not annotated and the agent learns jointly from all these demonstrations without knowing what each demonstration accomplishes. For the Kitchen environment we train the policy on top of R3M image representations following Nair et al. (2022b) using all demonstrations. This environment is challenging as it requires to distinguish which task is the one intended by the user among very similar opening tasks.

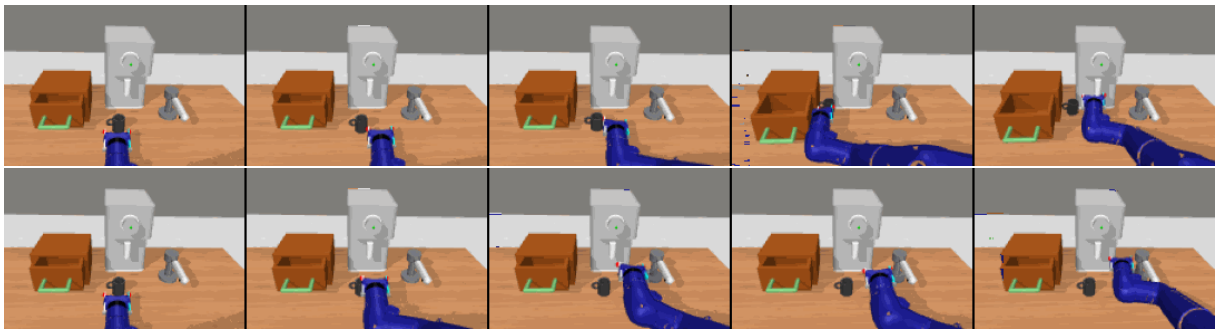
Following Chen et al. (2021a), for each of these tasks, we consider three different human video instructions collected in diverse environments to prompt our method at inference. Figure 4.7 illustrates the human video prompts used in the TableTop environments whereas Figure 4.8 shows the video prompts used in the Kitchen environments. Note that, in all environments, our approach uses the same video encoder and similarity metric.

4.4.2 Comparison to prior works

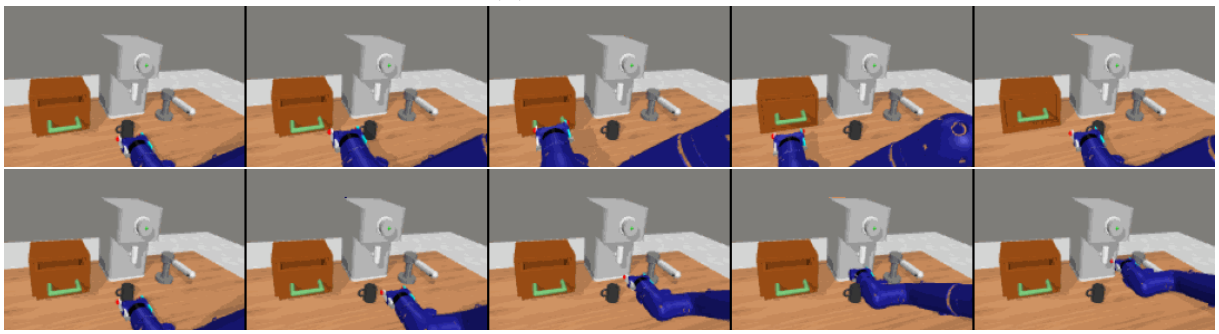
We first validate our video-conditioned control policy and compare it to the planning method of DVD (Chen et al., 2021a) on the TableTop environments in the following



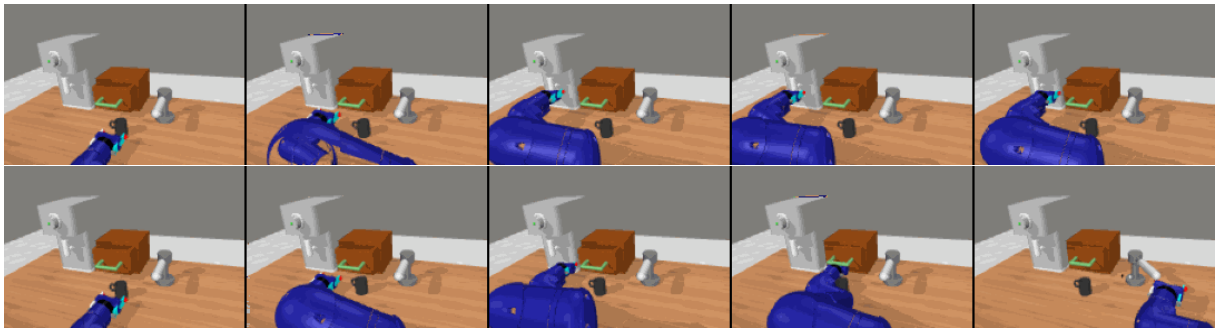
(a) Environment 1



(b) Environment 2



(c) Environment 3



(d) Environment 4

Figure 4.5 – Illustrations of random trajectories generated to train the policies in the TableTop environments.

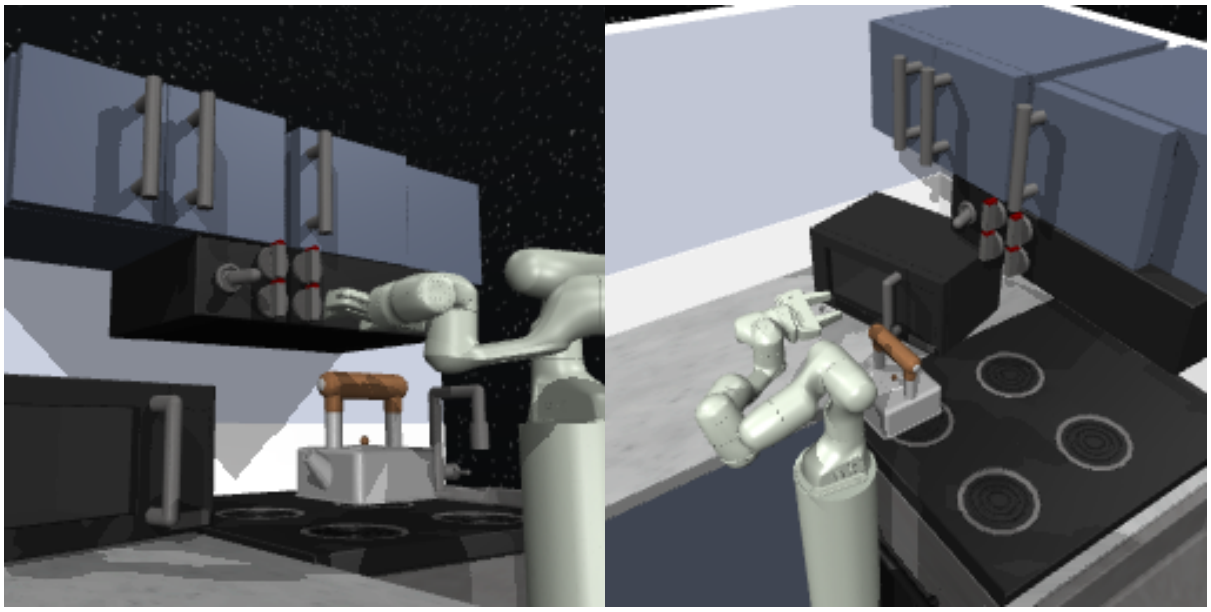


Figure 4.6 – Illustrations of Kitchen environment used in our experiments. Left: Kitchen left view. Right: Kitchen right view.

settings:

— *Seen robot demos*: the video similarity is trained on a subset of task-related SSv2 classes and on labelled videos of robot demonstrations for the target tasks collected in env 1 and the rearranged version of env 1,

— *Unseen robot demos*: the video similarity is trained on a subset of task-related SSv2 classes and on labelled videos of robot demonstrations corresponding to non-target tasks in env 1 and the rearranged version of env 1.

We compare our approach to Chen et al. (2021a) that uses the DVD similarity on top of a learned action-conditioned video prediction model for control (Babaeizadeh et al., 2017). For the *Seen robot demos* setting we take results reported in Chen et al. (2021a). For the *Uneen robot demos*, we take the best success rate reported in Chen et al. (2021a) for environment 1 and run the code of the authors to obtain results for environments 2, 3 and 4 (marked with *). For a fair comparison of our control policy, we here run ViP using the same video similarity function as in DVD which was trained on paired data between human videos and robot demonstrations.

Table 4.1 presents results for the TableTop environments where, for each environment, we report the average success rates across the three tasks and three human video instructions. ViP control policy outperforms DVD in both settings and most environments. We believe these improvements are due to comparing the human video prompt to full robot trajectories instead of shorter sub-trajectories and not relying on synthetic video generation for control as in Chen et al. (2021a). To demonstrate stability of our approach,

(a) Close drawer

(b) Push cup

(c) Turn faucet handle to the right

Figure 4.7 – Illustrations of the human video prompts used for the TableTop environments.

(a) Open left door

(b) Open sliding door

(c) Open microwave

Figure 4.8 – Illustrations of the human video prompts used in the kitchen environment.

Table 4.1 – Results for the TableTop environment using robot demonstrations. DVD results are obtained from Chen et al. (2021a) except the ones marked with "*" for which we run the code provided by the authors.

Method	env 1	env 2	env 3	env 4	Avg
DVD (Chen et al., 2021a)	65.2 (4.1)	62.3 (2.8)	53.8 (3.9)	**39.6** (0.7)	55.2 (2.9)
ViP (DVD similarity)	**97.1** (2.8)	**72.0** (15.9)	**82.1** (16.4)	**42.0** (11.2)	**73.3** (11.6)

(a) Seen robot demos setting

Method	env 1	env 2	env 3	env 4	Avg
DVD (Chen et al., 2021a)	55.1 (2.0)	**51.1*** (2.5)	38.4* (1.6)	35.0* (1.9)	44.9* (2.0)
ViP (DVD similarity)	**68.2** (11.2)	**55.2** (13.0)	**60.3** (8.0)	**43.6** (8.0)	**56.8** (10.0)

(b) Unseen robot demos setting

we report results over four training seeds while Chen et al. (2021a) reports results over evaluation runs.

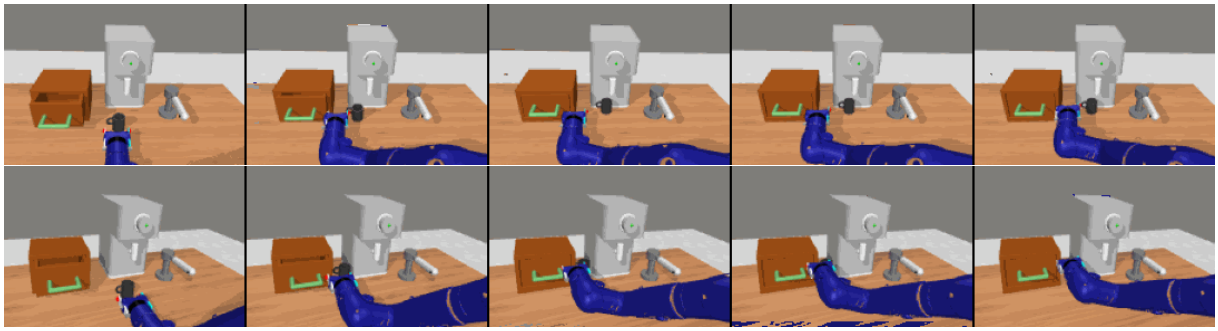
Moreover, our method is significantly faster than DVD as it operates in the space of pre-computed embeddings, while DVD requires to execute its video prediction model for each sampled trajectory. As result, in our experiments ViP requires less than 1 second to complete an episode in the TableTop environment while it takes more than 16 seconds to complete the same task with DVD.

We observe that the gains obtained by ViP compared to DVD are lower in the *unseen robot demos* setting. This can be attributed to the fact that our approach relies more on human videos of the target tasks to achieve good results.

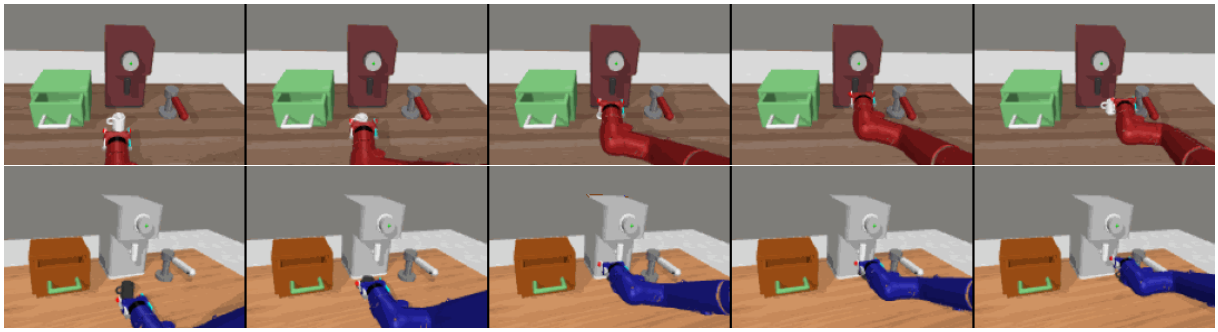
4.4.3 ViP without paired data

While DVD uses paired data with robot and human videos demonstrating execution of similar tasks, obtaining such data is cumbersome at scale, see discussion in Section 4.1. In this section we demonstrate that ViP is able to cope with a more challenging setting where the training of video similarity is done using pairs of human videos only and without access to any robot data. More precisely, we train on the same subset of SSV2 videos as the *Seen robot demos* setting of Section 4.4.2 without including any robot videos. This subset corresponds to the following SSV2 class labels: Closing something, Opening something, Moving something away from the camera, Moving something towards the camera, Pushing something from left to right and Pushing something from right to left.

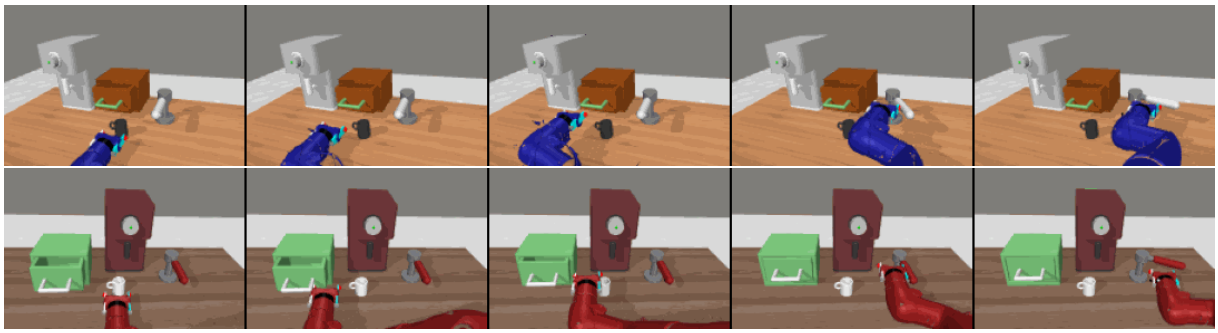
Table 4.2 presents our results. In this setting, our approach (*ViP (Ours)*) significantly outperforms DVD, succeeding to perform intended tasks more than 70% of the time



(a) Close the drawer



(b) Push the cup towards the coffee machine



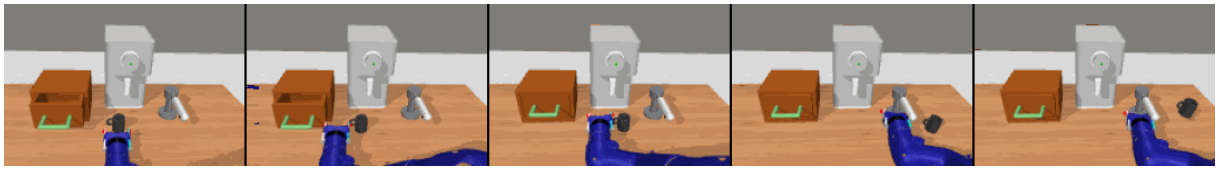
(c) Turn the faucet handle to the right

Figure 4.9 – Illustrations of successful rollouts in the TableTop environments given human video prompts.

Table 4.2 – Results for the TableTop environment without paired data. DVD results are obtained by running the code of the authors.

Method	env 1	env 2	env 3	env 4	Avg
Random	25.6	25.1	22.2	18.2	22.8
Human video as input	22.7 (7.9)	32.9 (12.9)	11.2 (7.8)	14.3 (9.6)	20.3 (9.6)
DVD(Chen et al., 2021a)	43.0* (2.6)	44.3* (2.0)	32.8* (1.7)	27.0* (2.6)	36.8* (2.2)
ViP (Cosine distance)	36.9 (4.5)	66.4 (17.5)	39.3 (15.0)	36.4 (15.4)	44.8 (13.1)
ViP (DVD similarity)	50.8 (10.2)	67.3 (17.4)	**72.6** (16.3)	**53.4** (9.6)	61.0 (13.4)
ViP (Ours)	**79.9** (11.4)	**77.9** (16.2)	**70.6** (17.8)	**56.9** (8.7)	**71.3** (13.5)

(a) Failure to close the drawer. Given a prompt of a human closing the drawer, the policy performs a pushing motion on the wrong object.



(b) Given a prompt of turning the faucet handle to the right, the policy pushes the cup to the right.

Figure 4.10 – Illustrations of failure cases in the TableTop environments.

on average across environments, while DVD performs only slightly better than chance. Moreover, we can see that our results in this setting are similar to results in Table 4.1 where paired robot demonstrations are used for similarity learning. Indeed, having a global understanding of full robot trajectories makes our action recognition approach less reliant on robot demos, which are more important when planning for short horizon as in DVD. Finally, ViP with the DVD similarity metric also performs well, showing that our approach can accommodate alternative choices of video similarity.

Figure 4.9 shows successful rollouts of our policy given human video prompts for the target tasks we evaluate on. Even though the policies were not trained on labelled robot videos and have thus no prior comprehension of what the expected outcomes are, the robot succeeds in performing the correct task. Note, however, that it may interact with undesired object. For instance, in Figure 4.9a, the robot correctly closes the drawer but also pushes the cup while performing the manipulation motion.

Figure 4.10 illustrates failure cases of our approach. Although the robot arm may execute the appropriate semantic movement, it may fail to manipulate the intended target object. In Figure 4.10b for instance, given a video of a human hand turning a faucet by moving from left to right (Figure 4.7c), the robot arm also moves from left to right, but instead of turning the faucet handle, it pushes the cup.

We further compare our approach to a baseline where we bypass our translation module and directly use human video embeddings to condition the ViP policy (*Human video as input*). Since ViP was only trained on robot video embeddings, embeddings of human videos are out of its training distribution. As result, the policy fails to perform the intended tasks. This highlights the importance of our translation procedure which matches human videos to a library of robot demonstrations.

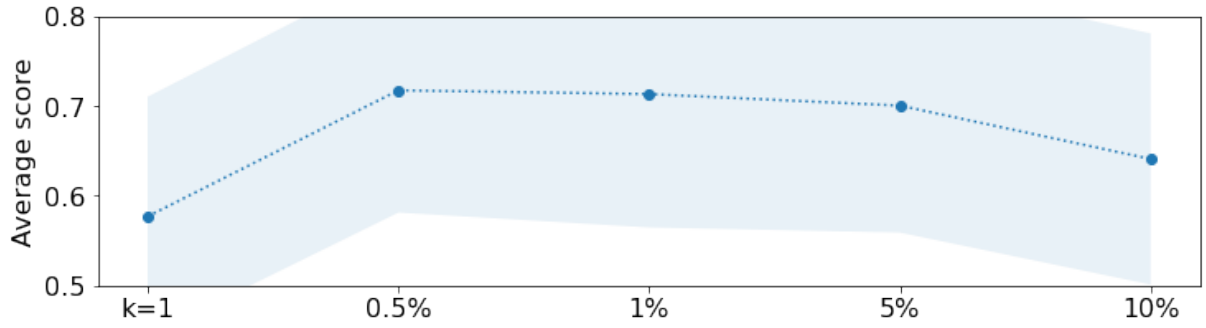


Figure 4.11 – Average success rate achieved by the policy on the TableTop environments for different values of k. We compare $k = 1$ against values of k corresponding to different percentages of the library size.

In Figure 4.11 we evaluate the sensitivity of our method to the number of nearest neighbor samples k used for translation of human videos in Section 4.3.3. High values of k ensure that the generated embedding e^r is an average of many appropriate robot embeddings, whereas low values of k prioritize maximising the similarity score between the predicted robot embedding and the human instruction embedding. While our approach is relatively robust to this hyperparameter, too high values of k result in lower performance. On the other side, for $k = 1$, meaning that we choose a robot demonstration with the highest similarity to the human video, the performance drops. We hypothesise that performing a nearest-neighbor regression has a regularization effect, as naively maximizing the similarity might result in choosing an adversarial robot embedding that is misleadingly considered appropriate for the target task. We set k to 1% of the library size for the TableTop environments and to 0.5% for the Kitchen environment.

In Table 4.2 we also ablate an alternative choice for video similarity where video embeddings obtained from the SSv2 action classification network are directly compared using a cosine distance measure *(Cosine distance)*. Using such a naive video similarity together with ViP results in superior performance compared to DVD, however, it is outperformed by both the DVD similarity with ViP (*ViP (DVD similarity)*) and our full approach, highlighting the importance of training an explicit distance between videos for the success of our method.

4.4.4 Kitchen environment

We evaluate our approach on the kitchen environments in the zero-shot setting. We compare our approach against a version of our method where we select a robot demonstration that solves the target task (*ViP (Oracle)*) as well as a version that randomly selects robot demonstrations from the library (*ViP (Random)*). We also compare against single-task R3M (Nair et al., 2022b), where we trained specialized policies for each task

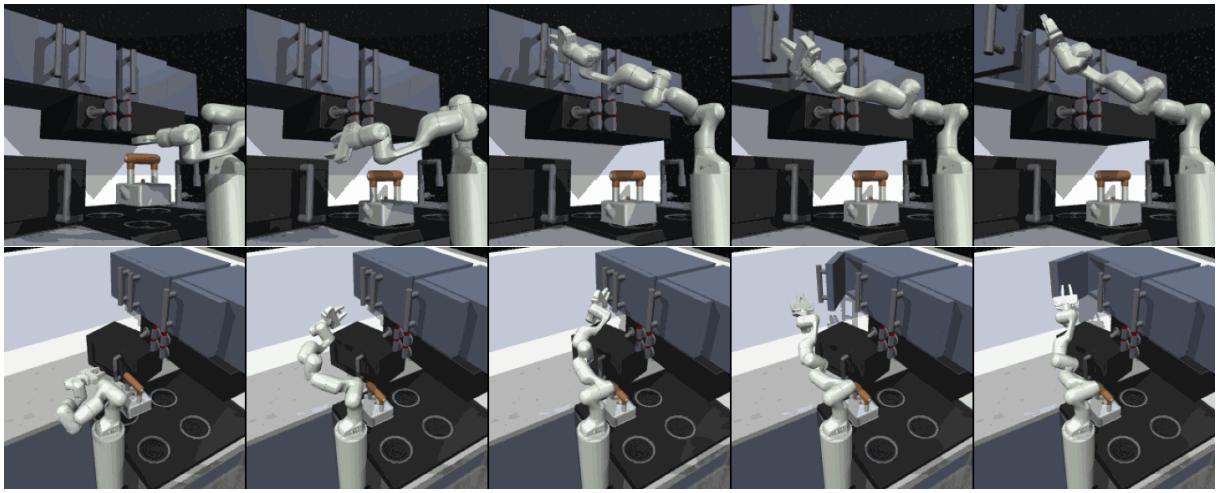
using all the demonstrations available.

We report the success rates for each task averaged over the 3 corresponding human video instructions across 4 training seeds in Table 4.3. Comparison between R3M and *Oracle* shows that our approach successfully learns from multi-task data for precise manipulation skills, achieving competitive results to R3M trained for single tasks. When prompted with a human video instruction in the left camera view (L), our approach successfully opens the correct door when presented with human videos of opening common doors and microwaves. However, the method fails to map human videos of opening sliding doors to appropriate robot skills. On the right camera view (R), our approach struggles more, often opening the wrong object when prompted with videos of sliding doors and microwaves.

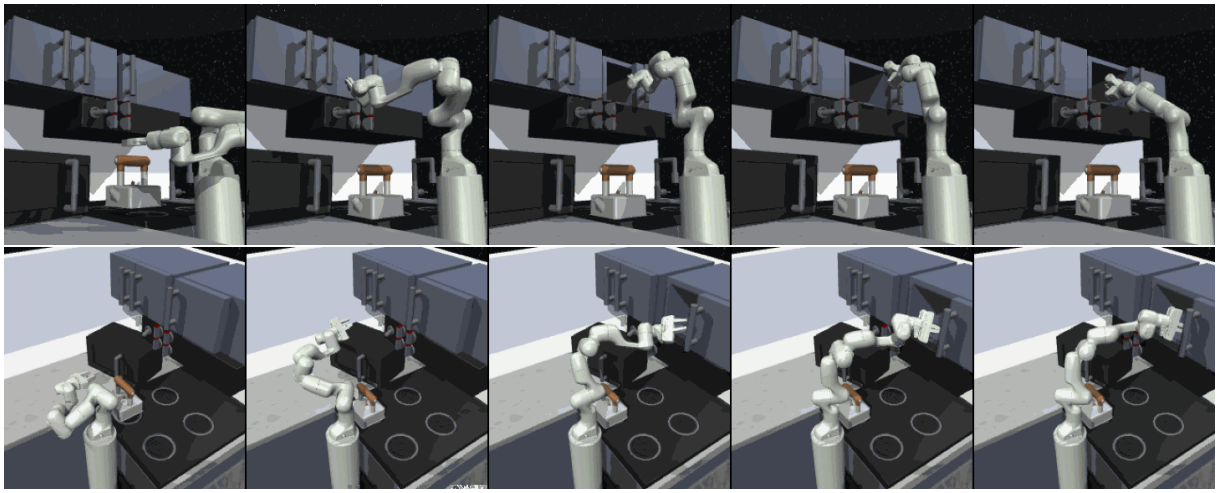
Figure 4.12 highlights successful rollouts of ViP. In these examples, the policy manages to reproduce the correct sequence of actions to solve the intended task given human video prompts, despite not having any additional knowledge or comprehension of what each training trajectory is actually accomplishing.

In Figure 4.13, we underline failure cases of our approach. First, similar to the Table-Top environments, the robot may perform the wrong manipulation task. In Figure 4.13a for instance, while the human prompt shows a human closing a sliding door, the robot chose to close the left door. This shows the limit of training on SSv2, where different human videos of opening something are encouraged to be grouped together by our similarity metric disregarding manipulated objects.

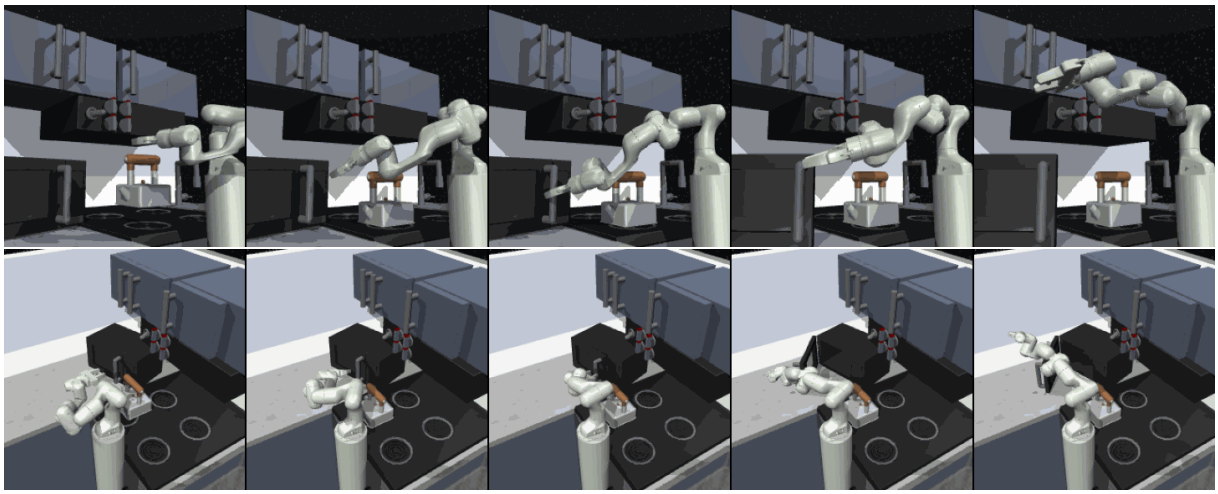
Moreover, in these kitchen environments that require precise manipulation skills, the policy may fail to correctly predict a sequence of actions to solve the task. Figure 4.13b shows such a failure case where, given a video of a human opening a microwave, the robot first moves the arm towards the microwave handle, then moves away from the object.



(a) Open left door

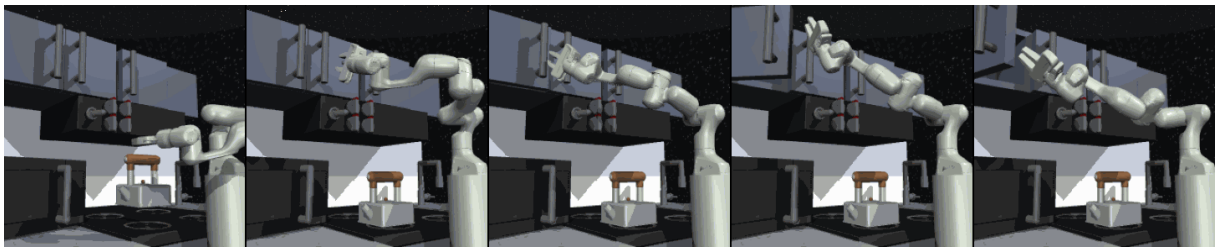


(b) Open sliding door

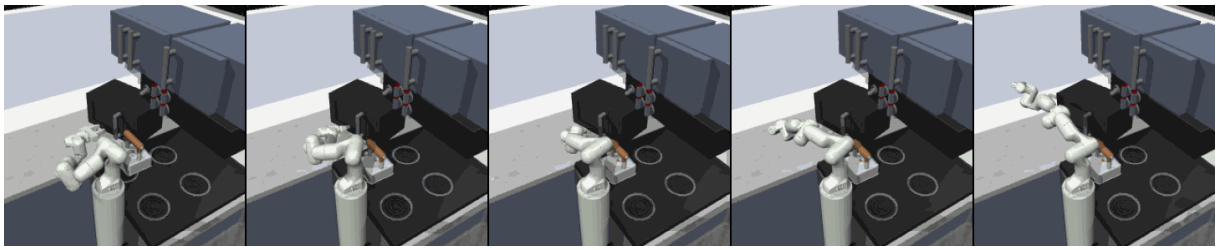


(c) Open microwave

Figure 4.12 – Illustrations of successful rollouts in the kitchen environments given human video prompts.



(a) Instead of opening the sliding door according to the human prompt, the robot opened the left door.



(b) The policy failed to predict a correct sequence of actions to solve the manipulation task of opening the microwave.

Figure 4.13 – Illustrations of failure cases in the kitchen environments.

Table 4.3 – Results on the Kitchen environment

Method	left door	sliding door	microwave
ViP (Random)	22.5 (2.7)	29.8 (0.4)	13.5 (4.7)
ViP (Oracle)	83.2 (16.5)	99.2 (0.8)	43.0 (10.2)
R3M (Nair et al., 2022b)	63.8 (10.3)	100 (0.0)	53.8 (11.0)
ViP (Ours)	69.4 (11.0)	41.7 (27.6)	25.1 (8.5)

(a) Left view

Method	left door	sliding door	microwave
ViP (Random)	22.0 (2.9)	30.0 (0.0)	8.5 (3.0)
ViP (Oracle)	92.0 (7.4)	99.8 (0.4)	27.2 (9.1)
R3M (Nair et al., 2022b)	57.5 (2.9)	100 (0.0)	30.0 (0.1)
ViP (Ours)	66.7 (3.3)	41.7 (14.4)	6.6 (2.9)

(b) Right view

4.5 Conclusion

We propose ViP, a method that learns to map human video instructions to robot skills in a zero-shot manner. We show that by conditioning on video embeddings, we can learn from multi-task robot data without supervision and prompt our policy with an unseen human video instruction at test time. As a step further towards less supervision in data-driven robotics, we demonstrate that, by training on large datasets of diverse labelled human videos, we don't need to pair human instructions to robot data during training.

Our experiments demonstrate that ViP can accommodate many different similarity metrics between human instructions and robot manipulations. Future work could explore other forms of similarities, such as mapping language instructions to robotic skills.

Chapter 5

Conclusion

In this last chapter, we summarize the contributions of this thesis in Section 5.1 before outlining challenges and directions for future work in Section 5.2.

5.1 Summary of contributions

This thesis has focused on learning multi-task control policies for robots with less expert supervision. One of our main objective was to reduce dependence on human supervision, given the inherent difficulties in manually collecting robot data. Our contributions are the following:

Learning goal-conditioned policies In Chapter 3, we present a novel goal-conditioned reinforcement learning algorithm for long-horizon reasoning. We leverage the compositionality of goal-reaching tasks to guide policy learning towards temporally extended goals through simpler subgoals imagined by a high-level policy. Unlike previous hierarchical approaches that generate subgoals at test time, our approach integrates subgoals into the training process, resulting in a policy that is simpler and faster to deploy at inference. Our method showcases superior success rates and sample efficiency when tackling intricate locomotion and vision-based manipulation tasks.

Following human video instructions In Chapter 4, we present a method that enables robot policies to follow human video instructions. Our approach relies neither on closely aligned human videos and robot settings nor on annotated robot demonstrations. Instead, we leverage large available datasets of videos to map video prompts to robot behavior in a zero-shot manner in multi-task vision based environments. With this approach, non-experts can easily communicate their desired task to the robot. We show that video encoders pretrained for human action recognition provide effective embedding spaces to

encode robot skills, allowing for generalization to tasks that the policy was not explicitly trained on.

5.2 Perspectives

As emphasized throughout this thesis, the scarcity of training data presents a significant challenge in the field of robot learning, impeding progress towards achieving general-purpose robotics. A straightforward approach to tackle this hurdle would involve collecting a massive amount of robot demonstrations and training behavior cloning policies at scale (Brohan et al., 2022). Although this avenue shows promise, it may not suffice to acquire internet-scale volume of robot demonstrations, as seen in various natural language processing and computer vision tasks. Hence, further advancements in online reinforcement learning and leveraging human data for robot learning, as we explored in our work, might be required to overcome these challenges. This section presents an overview of potential future directions that could be pursued following the research projects presented in this thesis.

Learning robot policies conditioned on natural image, video and language instructions In Chapter 3, we introduced a method for representing tasks as desired goals within the robot environment, which may not always be convenient for non-expert users to provide, whereas in Chapter 4, we prompt the policy using natural human videos. Moving forward, it would be valuable to explore how robot policies could be prompted using language instructions and natural images. Indeed, text-based instructions offer perhaps the most direct means for human operators to express their intentions, while many instructions with visual representations, such as Ikea assembly manuals, demonstrate desired outcomes through images captured in a different context than the robot environment. Additionally, considering instructions expressed in multiple modalities, such as captioned image instructions or captioned video clips, could further enhance the communication between humans and robots. Providing many ways to prompt the policy would empower human operators to best express their intentions based on the specific situation at hand (Figure 5.1). To achieve this, we could expand the method presented in Chapter 4 by training a similarity metric on datasets of captioned videos, such as Ego4D (Grauman et al., 2022) or Epic-Kitchens (Damen et al., 2018), using multimodal alignment techniques. This approach would allow for the creation of a shared embedding space that bridges different instruction modalities, enabling the alignment of text, image and video instructions to robot behaviors.

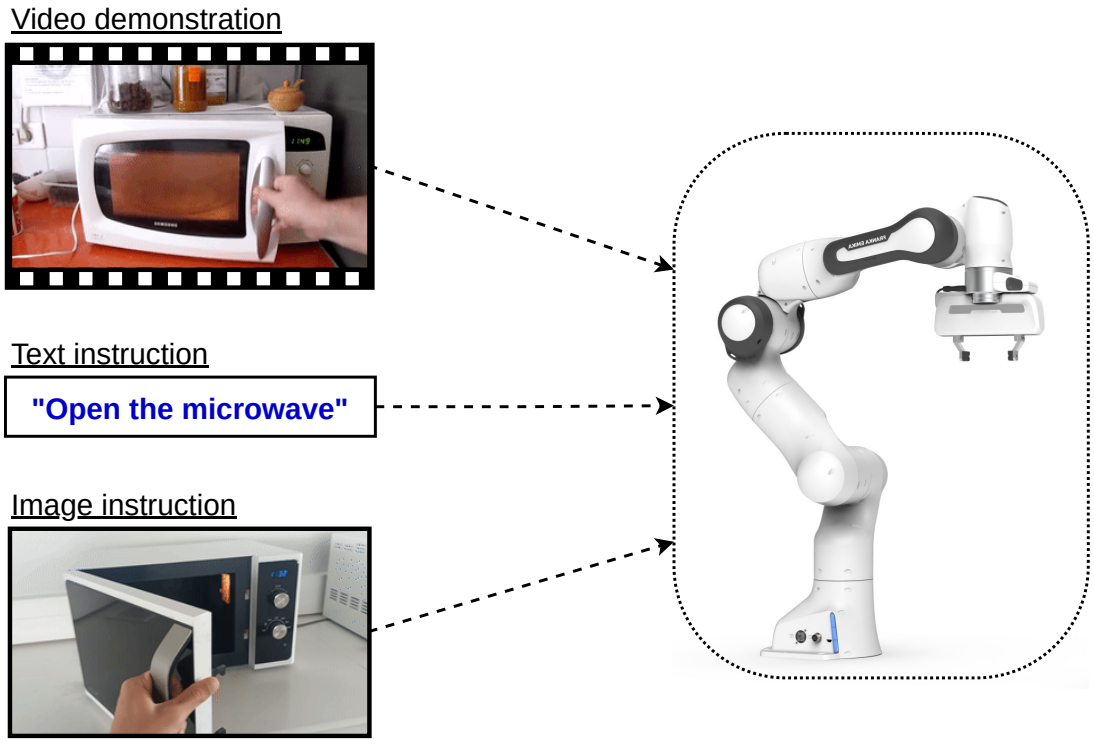


Figure 5.1 – A generalist robot should be able to understand human instructions expressed in various modalities, including video demonstrations, text instructions, or image-based depictions of the desired outcome.

Pretraining robot skills with large datasets of human videos The research outlined in Chapter 4 provides insights into obtaining pretrained representations for robot behaviors through the utilization of video action recognition techniques. Simultaneously, recent studies on universal image representations for robotics (Nair et al., 2022b; Ma et al., 2022; Xiao et al., 2022) suggest that large-scale videos offer a promising approach to effectively pretrain visual representations for robotics. This suggests that leveraging large-scale videos could be a fundamental factor in achieving successful pretraining for robot learning. Looking ahead, it would be valuable to devise a methodology that primarily leverages large video datasets for pretraining robot policies followed by a tiny amount of finetuning, either with a few robot demonstrations or with a small amount of online exploration, to embody the pretrained policy into the target robot. Such an approach, similar to the role of foundation models (Bommasani et al., 2021) in natural language processing and computer vision, has the potential to unlock truly generalizable robotics: once a policy has been pretrained, it could be seamlessly embodied into diverse robot platforms, accommodating variations in morphologies such as varying limb counts, wheel configurations, and sensor placements. Furthermore, this could allow robots to address tasks that were not covered by the collected robot demonstrations but were encountered in the pretraining videos.

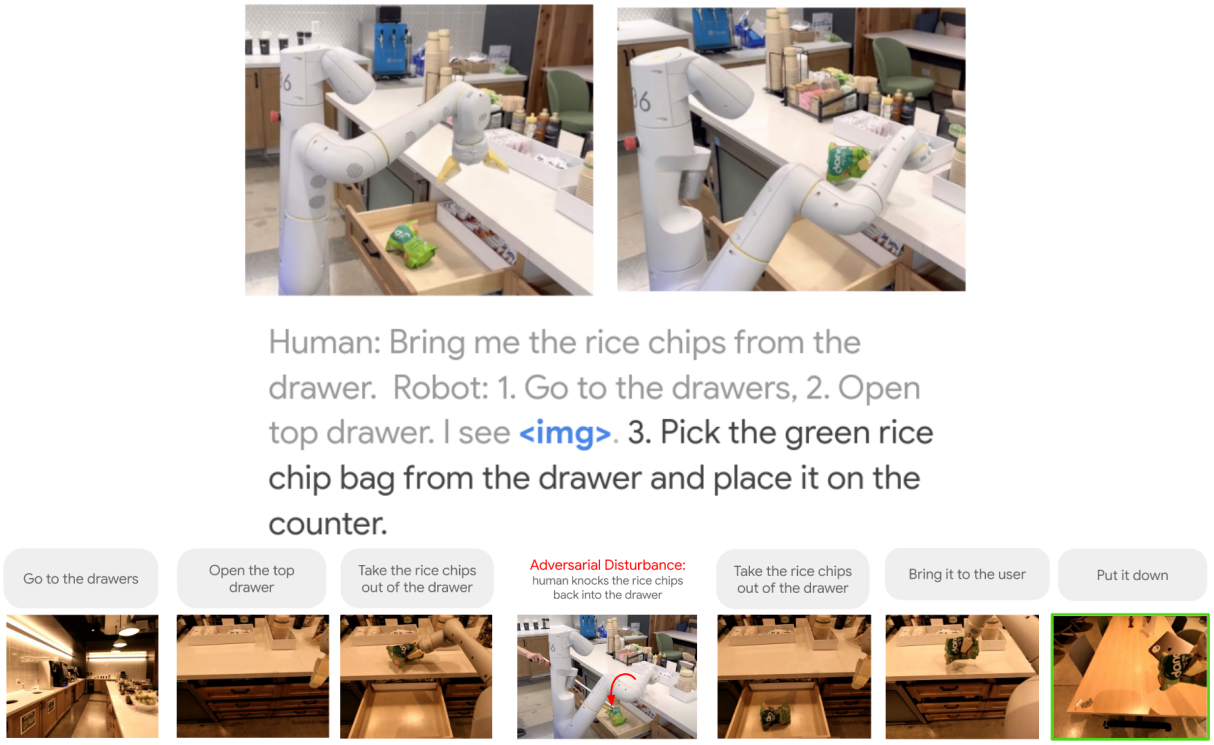


Figure 5.2 – Embodied language models, such as Palm-E (Driess et al., 2023), are able to generate plans for long-horizon instructions, which can subsequently be executed by lower-level navigation and manipulation policies.

Task planing with embodied language models Recent advance in large language models Brown et al. (2020) have demonstrated their remarkable capabilities to transform intricate language instructions into achievable plans of primitive commands that are directly executable by robots (Ahn et al., 2022; Liang et al., 2022; Driess et al., 2023; Shah et al., 2023; Singh et al., 2023). Large language models are currently one of the most promising candidates for high-level reasoning in robotics. This opens up a broad spectrum of opportunities for exploring how language models can be incorporated into real-world robots, allowing them to perform a variety of tasks. This entails investigating techniques to incorporate visual data, proprioceptive information, affordances, and safe interactions with humans or other robots within the framework of large language models (Figure 5.2). In this context, the method presented in Chapter 3 for aligning the high-level policy with the goal-conditioned policy offers a potential pathway to synchronize language models, serving as high-level controllers, with the capabilities of its low-level control policies.

Learning from unstructured play data In Chapter 3, we learn policies from scratch with data collected online whereas in Chapter 4 we consider structured demonstrations of fixed durations, acquired either randomly or from expert sources. However, a good way to gather data for robots is through "play" scenarios (Lynch et al., 2020), where operators

freely interact with the environment without a specific goal in mind. Such an approach is highly effective due to the potential to amass a larger amount f data compared to collecting demonstrations for each target tasks. Nevertheless, applying imitation learning techniques directly to such data poses challenges as it is unlabelled and lacks structure. Consequently, a promising avenue for future research is to adapt the methodologies presented in this thesis to harness the power of this type of robot data, enabling the learning of policies from a substantially larger and more diverse dataset. The method developed in Chapter 3 could be tailored to learn long-horizon policies from unstructured demonstrations. To overcome the lack of annotations and learn instruction-conditioned policies, we could leverage the methods presented in Chapter 4.

Multi-task dexterous manipulations with multimodal policies Policies learned through behavior cloning or reinforcement learning typically employ unimodal Gaussian action distributions as parameters. In its simplest instantiation, behavior cloning involves fitting expert trajectories by minimizing the mean-squared error between predicted actions and ground truth actions. Although this formulation as a regression problem offers simplicity, it suffers from a significant limitation: it can only accommodate unimodal distributions. This limitation becomes problematic for tasks that involve multimodality or learn from data collected by human operators that are inherently noisy, suboptimal and multimodal. Recent research has demonstrated that employing multimodal policies (Shafiullah et al., 2022; Chi et al., 2023; Zhao et al., 2023; Florence et al., 2022) allows for more precise manipulations while requiring fewer demonstrations. Extending these techniques to the multi-task setting, such as by incorporating multimodal policies instead of the behavior cloning component of the method described in Chapter 4, could be an intriguing avenue.

Bibliography

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

Abdolmaleki, A., Springenberg, J. T., Degrave, J., Bohez, S., Tassa, Y., Belov, D., Heess, N., and Riedmiller, M. (2018a). Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. A. (2018b). Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*.

Aghasadeghi, N. and Bretl, T. (2011). Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1561–1566. IEEE.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., et al. (2022). Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. (2019). Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Alakuijala, M., Dulac-Arnold, G., Mairal, J., Ponce, J., and Schmid, C. (2023). Learning reward functions for robotic manipulation by observing humans. *2023 IEEE International Conference on Robotics and Automation (ICRA)*.

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. (2017). Hindsight experience replay. In *Advances in Neural Information Processing Systems*.

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. (2017). Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*.

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*.

Bahl, S., Gupta, A., and Pathak, D. (2022). Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*.

Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. (2022). Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654.

Bardes, A., Ponce, J., and LeCun, Y. (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34–37.

Bennetts, J., Vardanega, P., Taylor, C., and Denton, S. (2016). Bridge data—what do we collect and how do we use it? In *Transforming the Future of Infrastructure through Smarter Information: Proceedings of the International Conference on Smart Infrastructure and ConstructionConstruction, 27–29 June 2016*, pages 531–536. ICE Publishing.

Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., and Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bonardi, A., James, S., and Davison, A. J. (2020). Learning one-shot imitation from humans without humans. *IEEE Robotics and Automation Letters*, 5(2):3533–3539.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. (2022). Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chabal, T., Strudel, R., Arlaud, E., Ponce, J., and Schmid, C. (2022). Assembly planning from observations under physical constraints. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10223–10229. IEEE.

Chane-Sane, E., Schmid, C., and Laptev, I. (2021). Goal-conditioned reinforcement learning with imagined subgoals. In *ICML*.

Chane-Sane, E., Schmid, C., and Laptev, I. (2023). Learning video-conditioned policies for unseen manipulation tasks. In *ICRA*.

Chebotar, Y., Hausman, K., Lu, Y., Xiao, T., Kalashnikov, D., Varley, J., Irpan, A., Eysenbach, B., Julian, R., Finn, C., et al. (2021). Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*.

Chen, A. S., Nair, S., and Finn, C. (2021a). Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021b). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, Z. and Huang, X. (2017). End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1856–1860. IEEE.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. (2023). Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*.

Choi, C., Taguchi, Y., Tuzel, O., Liu, M.-Y., and Ramalingam, S. (2012). Voting-based pose estimation for robotic assembly using a 3d sensor. In *2012 IEEE International Conference on Robotics and Automation*, pages 1724–1731. IEEE.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736.

Das, N., Bechtle, S., Davchev, T., Jayaraman, D., Rai, A., and Meier, F. (2020). Model-based inverse reinforcement learning from visual demonstrations. *arXiv preprint arXiv:2010.09034*.

Dayan, P. and Hinton, G. E. (1993). Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dietterich, T. G. (2000). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13.

Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. (2023). Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Edwards, A. D. and Isbell, C. L. (2019). Perceptual values from observation. *arXiv preprint arXiv:1905.07861*.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. (2021). Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*.

Erez, T., Tassa, Y., and Todorov, E. (2015). Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4397–4404. IEEE.

Eysenbach, B., Salakhutdinov, R., and Levine, S. (2019). Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems*.

Eysenbach, B., Salakhutdinov, R., and Levine, S. (2020a). C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

Eysenbach, B., Salakhutdinov, R., and Levine, S. (2020b). C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. (2022). Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*.

Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., and Tompson, J. (2022). Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR.

Florensa, C., Duan, Y., and Abbeel, P. (2017). Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*.

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. (2017). Meta learning shared hierarchies. *arXiv preprint arXiv:1710.09767*.

Fu, J., Luo, K., and Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*.

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*.

Furuta, H., Matsuo, Y., and Gu, S. S. (2021). Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*.

Galashov, A., Jayakumar, S. M., Hasenclever, L., Tirumala, D., Schwarz, J., Desjardins, G., Czarnecki, W. M., Teh, Y. W., Pascanu, R., and Heess, N. (2019). Information asymmetry in kl-regularized RL. In *International Conference on Learning Representations*.

Ghasemipour, S. K. S., Zemel, R., and Gu, S. (2020). A divergence minimization perspective on imitation learning methods. In Kaelbling, L. P., Kragic, D., and Sugiura, K., editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1259–1277. PMLR.

Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C., Eysenbach, B., and Levine, S. (2019). Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*.

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al. (2017). The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. (2022). Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.

Guhur, P.-L., Chen, S., Pinel, R. G., Tapaswi, M., Laptev, I., and Schmid, C. (2023). Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. (2019a). Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. (2019b). Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*.

Haarnoja, T., Hartikainen, K., Abbeel, P., and Levine, S. (2018a). Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 1851–1860. PMLR.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018b). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018c). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. (2018). Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heess, N., Wayne, G., Silver, D., Lillicrap, T. P., Erez, T., and Tassa, Y. (2015). Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Hong, Z.-W., Yu-Ming, C., Su, S.-Y., Shann, T.-Y., Chang, Y.-H., Yang, H.-K., Ho, B. H.-L., Tu, C.-C., Chang, Y.-C., Hsiao, T.-C., et al. (2018). Virtual-to-real: Learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. (2022). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.

Ichnowski, J., Avigal, Y., Kerr, J., and Goldberg, K. (2021). Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life: Third European Conference on Artificial Life Granada, Spain, June 4–6, 1995 Proceedings 3*, pages 704–720. Springer.

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. (2022). Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR.

Janner, M., Li, Q., and Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Jurgenson, T., Avner, O., Groshev, E., and Tamar, A. (2020). Sub-goal trees a framework for goal-based reinforcement learning. In *International Conference on Machine Learning*.

Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018). Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*.

Kappen, H. J., Gómez, V., and Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182.

Karamcheti, S., Nair, S., Chen, A. S., Kollar, T., Finn, C., Sadigh, D., and Liang, P. (2023). Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ke, L., Choudhury, S., Barnes, M., Sun, W., Lee, G., and Srinivasa, S. (2021). Imitation learning as f-divergence minimization. In *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pages 313–329. Springer.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

Koenemann, J., Del Prete, A., Tassa, Y., Todorov, E., Stasse, O., Bennewitz, M., and Mansard, N. (2015). Whole-body model-predictive control applied to the hrp-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351. IEEE.

Kostrikov, I., Nair, A., and Levine, S. (2021). Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Kostrikov, I., Yarats, D., and Fergus, R. (2020). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. (2019a). Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*.

Kumar, A., Hong, J., Singh, A., and Levine, S. (2022a). When should we prefer offline reinforcement learning over behavioral cloning? *arXiv preprint arXiv:2204.05618*.

Kumar, A., Peng, X. B., and Levine, S. (2019b). Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*.

Kumar, A., Singh, A., Ebert, F., Yang, Y., Finn, C., and Levine, S. (2022b). Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191.

Labbé, Y., Carpentier, J., Aubry, M., and Sivic, J. (2020). Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer.

Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D., and Sivic, J. (2022). Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. (2020). Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986.

Lee, Y., Sun, S.-H., Somasundaram, S., Hu, E. S., and Lim, J. J. (2019). Composing complex skills by learning transition policies. In *International Conference on Learning Representations*.

Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Levine, S. and Abbeel, P. (2014). Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Levine, S. and Koltun, V. (2013). Guided policy search. In *International Conference on Machine Learning*.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436.

Levy, A., Konidaris, G. D., Jr., R. P., and Saenko, K. (2019). Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*.

Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al. (2023). Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. (2022). Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*.

Lidec, Q. L., Jallet, W., Montaut, L., Laptev, I., Schmid, C., and Carpentier, J. (2023). Contact models in robotics: a comparative analysis. *arXiv preprint arXiv:2304.06372*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Litvak, Y., Biess, A., and Bar-Hillel, A. (2019). Learning pose estimation for high-precision robotic assembly using simulated depth images. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3521–3527. IEEE.

Liu, H., Lee, L., Lee, K., and Abbeel, P. (2022). Instruction-following agents with jointly pre-trained vision-language models. *arXiv preprint arXiv:2210.13431*.

Liu, H., Liu, G., Zhang, Y., Lei, L., Xie, H., Li, Y., and Sun, S. (2021). A 3d keypoints voting network for 6dof pose estimation in indoor scene. *Machines*, 9(10):230.

Liu, Y., Gupta, A., Abbeel, P., and Levine, S. (2018). Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lu, Y., Hausman, K., Chebotar, Y., Yan, M., Jang, E., Herzog, A., Xiao, T., Irpan, A., Khansari, M., Kalashnikov, D., et al. (2022). Aw-opt: Learning robotic skills with imitation andreinforcement at scale. In *Conference on Robot Learning*, pages 1078–1088. PMLR.

Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. (2020). Learning latent plans from play. In *Conference on Robot Learning*.

Lynch, C. and Sermanet, P. (2020). Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*.

Ma, Y. J., Liang, W., Som, V., Kumar, V., Zhang, A., Bastani, O., and Jayaraman, D. (2023). Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. (2022). Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Abbeel, P., Malik, J., et al. (2023). Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*.

Manuelli, L., Gao, W., Florence, P., and Tedrake, R. (2019). kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer.

Mees, O., Hermann, L., and Burgard, W. (2022a). What matters in language conditioned robotic imitation learning. *arXiv preprint arXiv:2204.06252*.

Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. (2022b). Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*.

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019). Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*.

Nagabandi, A., Konolige, K., Levine, S., and Kumar, V. (2020). Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR.

Nair, A., Dalal, M., Gupta, A., and Levine, S. (2020). Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*.

Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. (2018). Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*.

Nair, S. and Finn, C. (2020). Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*.

Nair, S., Mitchell, E., Chen, K., Savarese, S., Finn, C., et al. (2022a). Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. (2022b). R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*.

Nasiriany, S., Pong, V., Lin, S., and Levine, S. (2019). Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems*.

Ng, A. Y., Russell, S., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

Parascandolo, G., Buesing, L., Merel, J., Hasenclever, L., Aslanides, J., Hamrick, J. B., Heess, N., Neitz, A., and Weber, T. (2020). Divide-and-conquer monte carlo tree search for goal-directed planning. *arXiv preprint arXiv:2004.11410*.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. (2022). The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018a). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE.

Peng, X. B., Coumans, E., Zhang, T., Lee, T.-W., Tan, J., and Levine, S. (2020). Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*.

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. (2018b). Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6):1–14.

Pertsch, K., Lee, Y., and Lim, J. J. (2020a). Accelerating reinforcement learning with learned skill priors. *arXiv preprint arXiv:2010.11944*.

Pertsch, K., Rybkin, O., Ebert, F., Zhou, S., Jayaraman, D., Finn, C., and Levine, S. (2020b). Long-horizon visual planning with goal-conditioned hierarchical predictors. *Advances in Neural Information Processing Systems*.

Peters, J., Mülling, K., and Altun, Y. (2010). Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*.

Petrik, V., Tapaswi, M., Laptev, I., and Sivic, J. (2020). Learning object manipulation skills via approximate state estimation from real videos. *arXiv preprint arXiv:2011.06813*.

Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE.

Pirk, S., Khansari, M., Bai, Y., Lynch, C., and Sermanet, P. (2019). Online object representations with contrastive learning. *arXiv preprint arXiv:1906.04312*.

Pitis, S., Chan, H., Zhao, S., Stadie, B. C., and Ba, J. (2020). Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*.

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.

Pong, V., Gu, S., Dalal, M., and Levine, S. (2018). Temporal difference models: Model-free deep RL for model-based control. In *International Conference on Learning Representations*.

Rad, M. and Lepetit, V. (2017). Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3836.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., and Darrell, T. (2023). Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Raibert, M. H. (1986). *Legged robots that balance*. MIT press.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2012). On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems VIII*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2006). 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International journal of computer vision*, 66(3):231–259.

Rudin, N., Hoeller, D., Reist, P., and Hutter, M. (2022). Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International Conference on Machine Learning*.

Schmeckpeper, K., Rybkin, O., Daniilidis, K., Levine, S., and Finn, C. (2020a). Reinforcement learning with videos: Combining offline observations with interaction. *arXiv preprint arXiv:2011.06507*.

Schmeckpeper, K., Xie, A., Rybkin, O., Tian, S., Daniilidis, K., Levine, S., and Finn, C. (2020b). Learning predictive models from observation and interaction. In *European Conference on Computer Vision*, pages 708–725. Springer.

Schmidhuber, J. (2019). Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*.

Seo, Y., Lee, K., James, S. L., and Abbeel, P. (2022). Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. (2018). Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE.

Shafiullah, N. M., Cui, Z., Altanzaya, A. A., and Pinto, L. (2022). Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968.

Shah, D., Osiński, B., Levine, S., et al. (2023). Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pages 492–504. PMLR.

Shao, L., Migimatsu, T., Zhang, Q., Yang, K., and Bohg, J. (2021). Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434.

Sharma, M., Fantacci, C., Zhou, Y., Koppula, S., Heess, N., Scholz, J., and Aytar, Y. (2023). Lossless adaptation of pretrained vision models for robotic manipulation. In *The Eleventh International Conference on Learning Representations*.

Sharma, P., Pathak, D., and Gupta, A. (2019). Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32.

Shridhar, M., Manuelli, L., and Fox, D. (2022). Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR.

Siam, M., Elkerdawy, S., Jagersand, M., and Yogamani, S. (2017). Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 1–8. IEEE.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., and Riedmiller, M. (2020). Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. A. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning*.

Singh, A., Yu, A., Yang, J., Zhang, J., Kumar, A., and Levine, S. (2020). Cog: Connecting new skills to past experience with offline reinforcement learning. *arXiv preprint arXiv:2010.14500*.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. (2023). ProgPrompt: Generating situated robot task plans using large language models. In *International Conference on Robotics and Automation (ICRA)*.

Smith, L., Dhawan, N., Zhang, M., Abbeel, P., and Levine, S. (2019). Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*.

Smith, L., Kostrikov, I., and Levine, S. (2022). A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*.

Srivastava, R. K., Shyam, P., Mutz, F., Jaśkowski, W., and Schmidhuber, J. (2019). Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*.

Stevšić, S., Christen, S., and Hilliges, O. (2020). Learning to assemble: Estimating 6d poses for robotic object-object manipulation. *IEEE Robotics and Automation Letters*, 5(2):1159–1166.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7262–7272.

Strudel, R., Pashevich, A., Kalevatykh, I., Laptev, I., Sivic, J., and Schmid, C. (2020). Learning to combine primitive skills: A step towards versatile robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4637–4643. IEEE.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., Bohez, S., and Vanhoucke, V. (2018). Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*.

Teh, Y. W., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*.

Tirumala, D., Noh, H., Galashov, A., Hasenclever, L., Ahuja, A., Wayne, G., Pascanu, R., Teh, Y. W., and Heess, N. (2019). Exploiting hierarchy for learning and transfer in kl-regularized rl. *arXiv preprint arXiv:1903.07438*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE.

Toussaint, M. (2009). Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Veeriah, V., Oh, J., and Singh, S. (2018). Many-goals reinforcement learning. *arXiv preprint arXiv:1806.09605*.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*.

Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., and Savarese, S. (2019). Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352.

Wang, Z., Novikov, A., Zolna, K., Merel, J., Springenberg, J. T., Reed, S. E., Shahriari, B., Siegel, N. Y., Gülçehre, Ç., Heess, N., and de Freitas, N. (2020). Critic regularized regression. In *Advances in Neural Information Processing Systems*.

Wiering, M. and Schmidhuber, J. (1997). Hq-learning. *Adaptive Behavior*, 6(2):219–246.

Wu, P., Escontrela, A., Hafner, D., Abbeel, P., and Goldberg, K. (2023). Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pages 2226–2240. PMLR.

Wu, Y., Tucker, G., and Nachum, O. (2019). Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

Xiao, T., Radosavovic, I., Darrell, T., and Malik, J. (2022). Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*.

Xiong, H., Li, Q., Chen, Y.-C., Bharadhwaj, H., Sinha, S., and Garg, A. (2021). Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE.

Xu, T., Li, Z., and Yu, Y. (2020). Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33:15737–15749.

Yu, T., Finn, C., Xie, A., Dasari, S., Zhang, T., Abbeel, P., and Levine, S. (2018). One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*.

Zhang, L., Yang, G., and Stadie, B. C. (2020). World model as a graph: Learning latent landmarks for planning. *arXiv preprint arXiv:2011.12491*.

Zhang, T., Eysenbach, B., Salakhutdinov, R., Levine, S., and Gonzalez, J. E. (2021). C-planning: An automatic curriculum for learning goal-reaching tasks. *arXiv preprint arXiv:2110.12080*.

Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., and Abbeel, P. (2018). Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE.

Zhao, R., Sun, X., and Tresp, V. (2019). Maximum entropy-regularized multi-goal reinforcement learning. In *International Conference on Machine Learning*.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. (2023). Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*.

Zorina, K., Carpentier, J., Sivic, J., and Petrík, V. (2021). Learning to manipulate tools by aligning simulation to video demonstration. *IEEE Robotics and Automation Letters*, 7(1):438–445.

Chapter 6

Résumé en Français

6.1 Introduction

Au cours de la dernière décennie, les progrès en apprentissage profond ont abouti à des avancées majeures dans de nombreux domaines de l'intelligence artificielle tel que la classification et la segmentation d'images, la détection d'objets ou la compréhension du langage. Néanmoins, l'apprentissage profond pour le contrôle robotique, c'est à dire la génération de mouvements en utilisant des approches basées sur le traitement de données, reste un défi majeur. Ainsi, les avancées en apprentissage profond n'ont pas encore été traduites par l'avènement de robots généralistes capables d'opérer de manière autonome dans des environnements complexes et dynamiques, limitant ainsi leurs déploiements dans le monde réel au delà des laboratoires et des chaînes de productions robotisées.

Le but de cette thèse est de proposer de nouveaux algorithmes d'apprentissage de politiques de contrôles robotiques multi-tâches. Il s'agit d'entraîner des réseaux de neurones qui génèrent, à partir des observations du robot telles que les images issues de caméras ou les signaux des capteurs de contacts et proprioceptifs, les commandes à envoyer aux actionneurs du robot à chaque pas de temps pour permettre la génération d'un mouvement qui effectue la tâche voulue par l'utilisateur. Dans cette thèse, nous nous concentrons sur trois aspects principaux : l'apprentissage multi-tâches, le raisonnement à long terme et la compréhension des instructions humaines.

Apprentissage de politiques multi-tâche L'apprentissage de politiques multi-tâches vise à former un seul réseau neuronal capable d'exécuter diverses tâches, créant ainsi des contrôleurs robotiques polyvalents. Cette approche tire parti des similarités entre les tâches, améliorant les performances du contrôleur et favorisant la généralisation à de nouvelles tâches sur lesquelles le robot n'a pas été explicitement entraîné. Cependant, cette méthode pose des défis, notamment la nécessité de paramétrer de nombreux mouvements,

et donc de nombreuses tâches, dans un seul réseau neuronal, ainsi que la transmission effective des objectifs de l'utilisateur au robot.

Raisonnement sur le long terme L'un des défis majeurs en robotique est la capacité à raisonner sur le long terme, impliquant la planification et l'exécution de tâches complexes s'étalant sur plusieurs étapes impliquant de nombreuses sous-tâches, telles que la préparation d'un repas ou l'assemblage d'un objet complexe. Ce type de raisonnement étendu dans la temporalité est souvent divisé en deux niveaux: un raisonnement de bas niveau (comprenant des compétences motrices de base) et un raisonnement de haut niveau (impliquant la planification pour atteindre des objectifs à long terme). Un des objectifs de cette thèse est donc également de proposer des algorithmes d'apprentissage permettant aux robots de gérer efficacement des tâches complexes en décomposant hiérarchiquement les compétences nécessaires, sans recourir à une supervision humaine supplémentaire.

Suivre des vidéos de démonstration humaine Un autre objectif de notre thèse est de pouvoir définir les tâches à travers des instructions vidéo humaines que le robot est amener à imiter. En effet, cette approche offre une solution pratique pour les utilisateurs non experts de spécifier la tâche voulue au robot. Nous sommes motivés par le potentiel d'utilisation de vastes ensembles de données vidéo en robotique pour améliorer l'apprentissage de nouvelles tâches, impliquant le fait que les vidéos peuvent avoir été collectées dans des contextes divers et naturels, comme on peut le trouver sur internet.

6.2 Apprentissage par renforcement avec des sous-goals imaginés

Dans la première contribution de cette thèse, nous présentons un nouvel algorithme d'apprentissage par renforcement pour atteindre des objectifs avec l'aide de sous-objectifs imaginés. Cette contribution a été présentée à l'*International Conference on Machine Learning (ICML) 2021.*

6.2.1 Problème

Nous nous intéressons au problème de l'apprentissage de politiques conditionnées à un objectif, une configuration que le robot doit atteindre dans l'environnement. Dans le cadre de la locomotion, il peut par exemple s'agir d'une position que le robot doit atteindre en générant la bonne séquence d'actions pour faire déplacer le robot à l'endroit voulu. Dans le contexte de la manipulation à partir d'images, il est par exemple possible

de communiquer la tâche voulue au robot sous la forme d'une image de la configuration de l'espace de travail du robot voulue par l'utilisateur. La politique doit alors générer une séquence d'actions pour faire bouger le bras manipulateur du robot jusqu'à ce que la configuration du robot et des objets de l'environnement corresponde à celle décrite par l'image.

Formuler les problèmes de contrôle robotique sous la forme de but à atteindre possède de nombreux avantages. En effet, de nombreuses tâches de locomotion et de manipulations peuvent être formulées comme une configuration à atteindre dans l'environnement du robot: se mouvoir dans toutes les directions, déplacer des objets, etc... Dans ce cadre, l'ensemble des configurations possibles offre tout autant de tâches que le robot peut être amené à accomplir. L'ensemble des configurations possibles constitue par ailleurs un espace continu au travers duquel la politique peut généraliser à de nouvelles tâches. Enfin, la fonction de récompense peut être très simplement définie: dans nos expériences, le robot reçoit une pénalité de -1 jusqu'à ce qu'il atteigne l'objectif demandé, l'encourageant ainsi à accomplir la tâche le plus rapidement possible.

Bien que les approches par renforcement existantes permettent de résoudre ce problème pour des configurations simples à atteindre, ces méthodes peuvent parfois avoir des difficultés à atteindre des objectifs qui demandent une longue séquence d'actions à accomplir. Pour adresser ces problèmes de raisonnement à long horizon, il est possible d'utiliser des méthodes hiérarchiques, dans lesquelles une politique de haut niveau module une politique de bas niveau en proposant une séquence de sous-objectifs menant à la résolution de la tâche. Cependant, entraîner et déployer des politiques hiérarchiques ajoute des problèmes d'instabilité et rend le déploiement de la politique plus coûteux qu'une politique simple.

6.2.2 Méthode

Nous proposons *RIS*, un nouvel algorithme d'apprentissage par renforcement hiérarchique pour atteindre des objectifs arbitraires dans l'environnement du robot. Comme de nombreuses approches hiérarchiques antérieures, nous proposons d'utiliser des sous-objectifs pour décomposer une tâche d'atteinte d'un objectif lointain en plusieurs sous-objectifs plus simples à atteindre de l'un à l'autre. Cependant, contrairement aux approches hiérarchiques antérieures, nous proposons d'utiliser les sous-objectifs pour faciliter et accélérer l'apprentissage de la politique pendant la phase d'entraînement, et non pas pendant le déploiement.

Le fonctionnement de notre approche est illustré dans la Figure 6.1. Supposons que nous souhaitons que le robot atteigne un objectif lointain g que la politique ne parvient pas encore à accomplir. Si nous avons accès à un sous-objectif s_g que la politique parvient

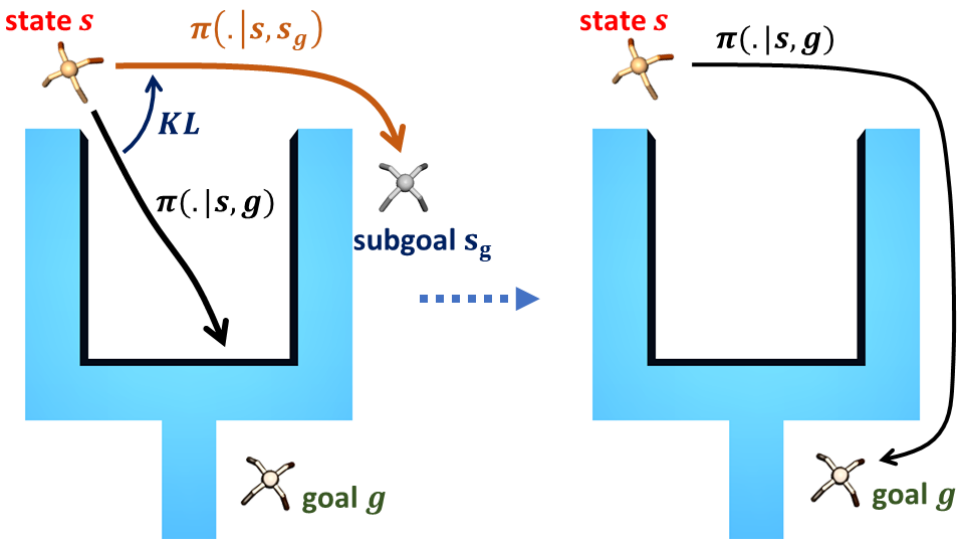
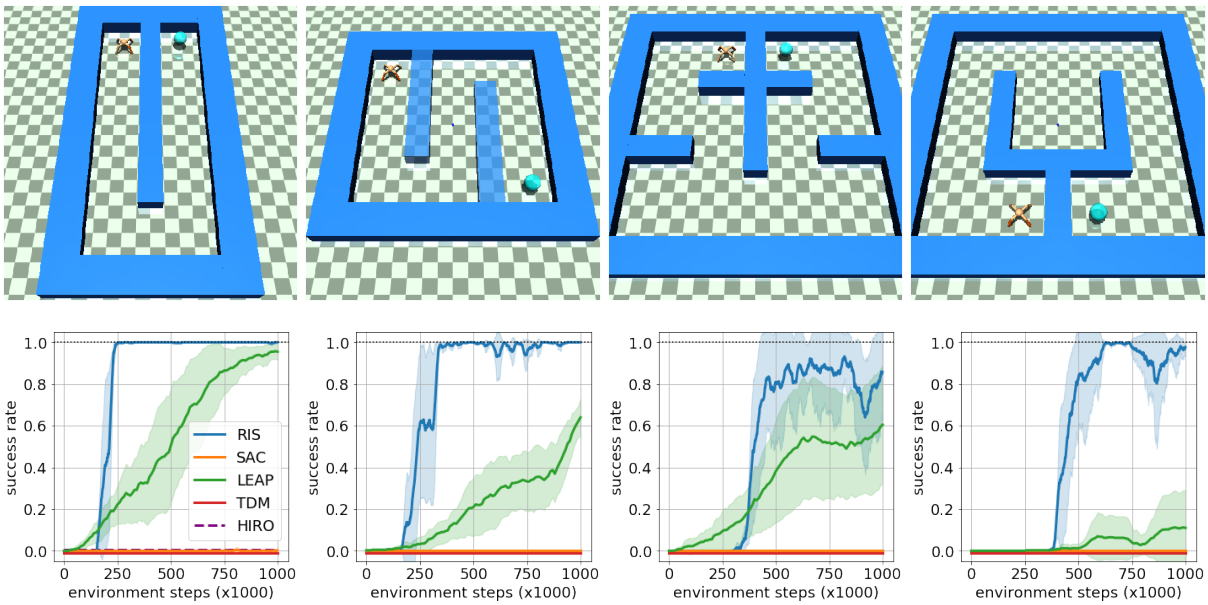


Figure 6.1 – Illustration de l'apprentissage de politique régularisé en utilisant des sous-objectifs imaginés. (À gauche) : La politique échoue à atteindre un objectif lointain, mais elle peut atteindre un sous-objectif plus proche. Notre approche génère automatiquement des sous-objectifs imaginés pour aider l'entraînement de la politique. (À droite) : Une fois l'entraînement terminé, la politique peut atteindre des objectifs arbitrairement éloignés sans avoir recours à des sous-objectifs.

déjà à accomplir depuis l'état s, alors nous pouvons nous attendre à ce que l'action à effectuer pour atteindre g soit similaire à celle pour atteindre s_g. Notre approche encourage cela pour toutes les configurations d'objectifs de l'environnement, en régularisant la politique avec la distance de Kullback-Leibler entre la solution qu'il cherche à obtenir pour atteindre un goal lointain g et les sous-goals possibles s_g obtenus par une politique de haut-niveau entraînée conjointement. A la fin de l'entraînement, la politique issue de cette entraînement est capable d'atteindre des objectifs arbitrairement lointains sans avoir recours à des sous-objectifs après déploiement. Puisque ces sous-objectifs ne sont utilisés que dans la fonction de coût de l'entraînement de la politique, mais qu'elles ne sont pas utilisées lors des phases d'interactions avec l'environnement du robot, nous les appelons des sous-objectifs imaginés.

Notre approche ne nécessite pas de prédire une séquence de sous-objectifs pour chaque objectif, un seul sous-objectif suffit. Nous avons donc également développé une méthode d'apprentissage de politique de haut niveau pour pouvoir prédire des sous-objectifs situés à mi-chemin de l'objectif final. Cette méthode utilise la fonction de valeur de la politique d'atteinte d'objectifs comme une mesure de distance entre les états. Cette distance évolue avec les capacités d'accomplissement d'objectifs de la politique pendant l'entraînement.

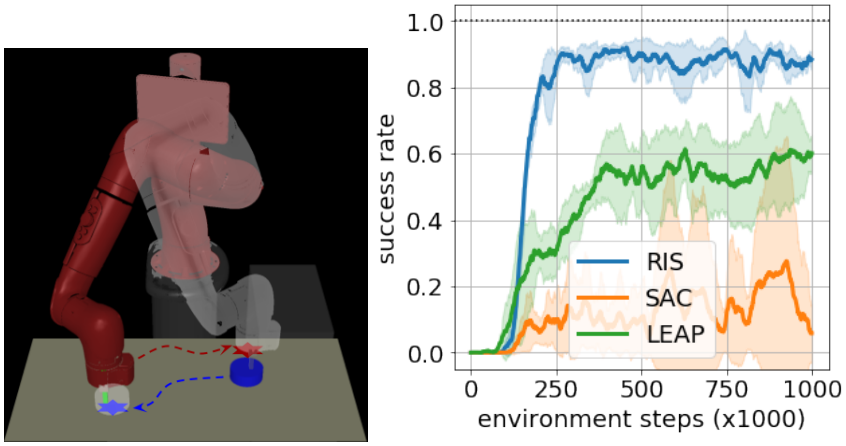


(a) Environnement "U" (b) Environnement "S" (c) Environnement "Π" (d) Environnement "ω"

Figure 6.2 – Comparaison de RIS avec des méthodes antérieures (rangée du bas) sur 4 tâches différentes de navigation de robot fourmis. Nous évaluons le taux de réussite de l'agent sur les configurations difficiles illustrées dans la rangée du haut, où la fourmi est située à l'état initial et l'emplacement de l'objectif souhaité est représenté par une sphère cyan.

6.2.3 Résultats expérimentaux

Locomotion dans des labyrinthes Nous évaluons notre approche sur des tâches complexes de locomotion de robots fourmis dans des labyrinthes en simulation, illustrées en Figure 6.2. La politique observe les informations issues de ses capteurs proprioceptifs et sa position dans l'environnement, sans connaître les positions des murs. Elle doit prédire la séquence des forces à appliquer sur les joints du robot de façon à atteindre l'objectif illustré par une sphère cyan. Pendant l'entraînement, le robot s'entraîne à atteindre n'importe quelle configuration depuis n'importe quel état initial. Pendant la phase d'évaluation, nous testons la politique dans la configuration la plus difficile offerte par l'environnement, demandant du raisonnement à long terme pour à la fois permettre les déplacements du robot tout en résolvant le labyrinthe. Comme illustré en Figure 6.2, notre approche atteint des taux de succès plus élevés que les approches antérieures tout en nécessitant moins d'interactions avec l'environnement. Par ailleurs, une fois entraînée, notre approche nécessite beaucoup moins de puissance de calcul pour déployer la politique contrairement à des approches comme LEAP.



(a) Illustration de l'environnement de manipulation robotique

(b) Comparison à l'etat de l'art

Figure 6.3 – Environnement de manipulation robotique: (a) illustration de la tâche; (b) comparaison de notre approche avec LEAP et SAC.

Manipulation à partir de la vison Nous évaluons également notre approche sur une tâche de manipulation à partir d'images, dans lequel le robot doit déplacer le palet ainsi que son bras pour atteindre la configuration donnée par une image, comme illustré en Figure 6.3. La politique observe l'image de sa configuration actuelle et contrôle la vélocité du bras robot sur la table. Comme l'espace d'observation du robot correspond à des images de couleurs de grande dimension, notre politique neuronale inclut également un encodeur d'image basé sur des couches convolutionnelles. La prédiction de sous-objectifs imaginés se fait dans l'espace latent de l'encodeur plutôt que dans l'espace des images. Évaluée sur des configurations complexes de l'environnement, notre approche atteint des taux de succès plus élevés que les approches antérieures tout en nécessitant moins d'interactions avec l'environnement, montrant l'efficacité de notre méthode sur des environnements de manipulation à partir d'images.

6.3 Apprentissage de politiques à partir de vidéos d'humains

Dans le chapitre précédent, nous avons introduit une méthode pour améliorer l'apprentissage par renforcement conditionné à un objectif pour les tâches de long horizon. Néanmoins, une définition des tâches basée uniquement sur des configurations d'objectifs à atteindre présente des limites. Dans la deuxième contribution de cette thèse, nous présentons un algorithme d'apprentissage de politiques capables d'imiter des vidéos de démonstrations

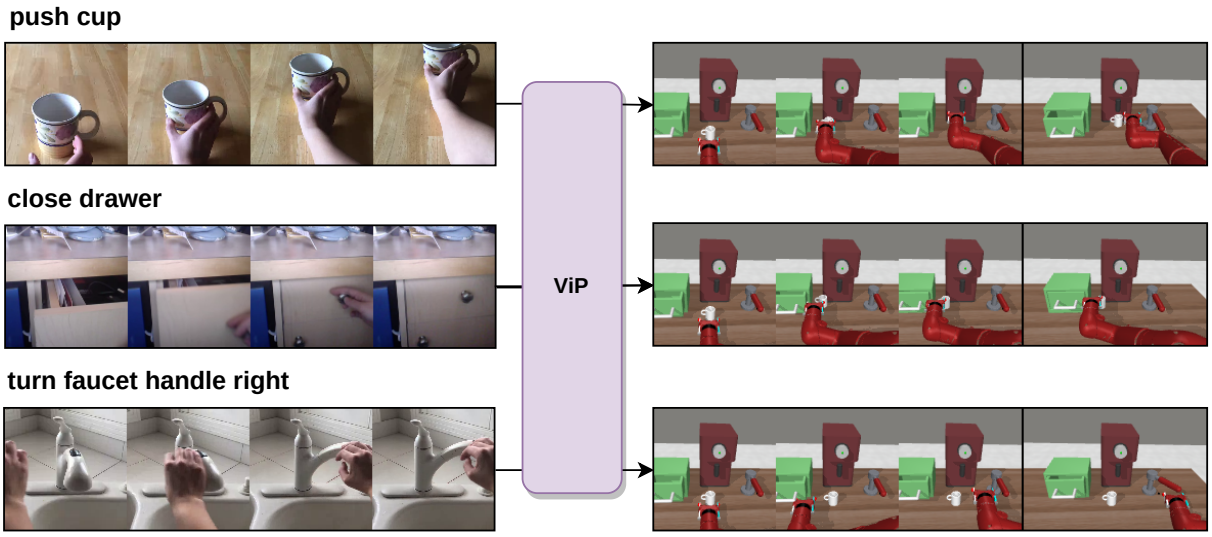


Figure 6.4 – Notre méthode ViP commande le robot à partir d'une vidéo de démonstration humaine capturée dans un environnement naturel dans le but de réaliser une tâche similaire à celle décrite par la vidéo. Étant donnée une instruction vidéo humaine dans une scène non robotique, notre politique conditionnée par la vidéo (ViP) contrôle le robot pour effectuer une tâche similaire sans entraînement préalable, c'est-à-dire que l'agent n'observe jamais de données de robot associées à des instructions humaines pendant l'entraînement et détermine quelles compétences de manipulation il est censé effectuer dans son environnement au moment du test. Nous illustrons trois exemples de différentes tâches démontrées par des personnes et les séquences générées correspondantes par notre méthode dans un environnement de simulation robotique.

humaines. Cette contribution a été présentée à l'*International Conference on Robotics and Automation (ICRA) 2023*.

6.3.1 Problème

Nous proposons *ViP*, un algorithme d'apprentissage de politiques visuelles de manipulation robotique conditionnée par la vidéo. Comme illustré en Figure 6.4, le but est de proposer un algorithme qui génère, à partir d'une instruction humaine sous la forme d'une vidéo de manipulation, le mouvement nécessaire pour reproduire la tâche correspondante dans l'environnement.

Dans ce contexte, l'utilisateur fournit ainsi une vidéo de la tâche de manipulation qu'il doit faire accomplir au robot. Cette vidéo peut avoir été enregistrée dans un environnement très diffèrent visuellement de celui dans lequel le robot évolue. Malgré cet écart de domaine entre vidéo humaine et environnement robotique, nous supposons que le robot ne connaît à priori pas ce qu'il est censé faire à partir de l'instruction vidéo. Par exemple, dans l'environnement 6.4, trois tâches sont à posteriori possibles, et donc nous

pouvons tester le robot sur trois vidéos humaines différentes, bien que la politique ne se soit pas explicitement entraînée sur ces vidéos.

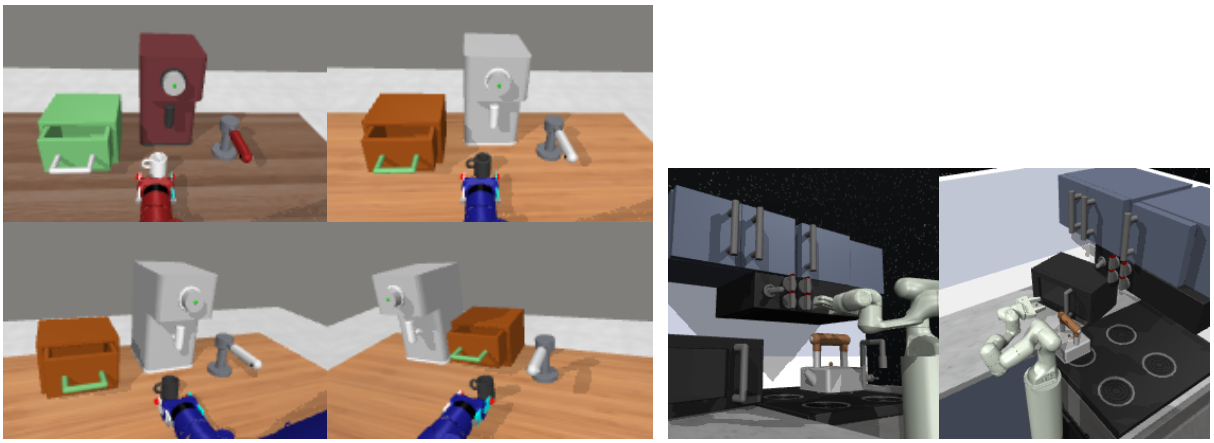
Ces hypothèses contrastent avec de nombreux travaux antérieurs à l'intersection de l'apprentissage robotique et de la compréhension vidéo qui supposent typiquement que le robot doit imiter une vidéo unique alignée avec l'environnement robotique, ce qui n'est souvent possible que si la vidéo est enregistrée minutieusement dans un laboratoire. Notre cadre de travail, plus réaliste, permet à l'utilisateur de communiquer la tâche qu'il souhaite voir effectuer par le robot de façon très simple et ouvre la possibilité d'apprendre des politiques de robotique à partir de vaste jeux de données de vidéo d'humains, comme on peut facilement en trouver sur internet.

6.3.2 Méthode

Notre algorithme opère en trois étapes: dans une première phase nous apprenons un encodeur de vidéos doté d'une mesure de similarité entre vidéos à partir d'un vaste jeux de données de vidéos de manipulations humaines. Dans une deuxième phase, nous apprenons une politique à générer des trajectoires robotiques à partir des vidéos de robot correspondantes. Lors de la phase de déploiement, nous utilisons la mesure de similarité pour trouver, à partir de l'instruction vidéo humaine, une vidéo de robot qui s'y rapporte et nous utilisons la politique pour générer le mouvement correspondant.

Apprentissage d'une mesure de similarité entre vidéos Nous utilisons des techniques d'apprentissage contrastif pour entraîner un encodeur de vidéos composé de couches de convolutions 3D à partir d'un sous-ensemble du vaste jeux de données *Something-Something-v2*. Ce jeu de données est constitué de milliers de vidéos de manipulations simples collectées par des humains dans des environnements variés. Les vidéos sont étiquetées avec l'action effectuée dans la vidéo, par exemple "pousser quelque chose" ou "ouvrir quelque chose". Notre apprentissage contrastif fait en sorte que l'encodeur produise des représentations de vidéos qui sont proches entre elles lorsqu'il s'agit de vidéos de la même tâche tandis que ces représentations doivent être éloignées pour des vidéos de tâches différentes. Bien que l'entraînement ne se fasse uniquement qu'à partir de vidéos humaines, grâce à la taille et la grande diversité de ce jeu de données, cette mesure de similarité généralise et est capable de comparer des vidéos de robots avec des vidéos d'humains.

Apprentissage supervisé de politiques conditionnées par des vidéos Dans cette deuxième étape, nous entraînons une politique neuronale à reproduire des mouvements à partir d'un jeu de données de trajectoires collectées dans l'environnement du robot. Ce



(a) Environnements "table de travail" (b) Environnements "cuisine"

Figure 6.5 – Illustrations des environnements "table de travail" à gauche et "cuisine" à droite utilisés lors de nos experiences.

jeu de données peut être collecté de différentes façons, mais nous supposons que nous ne savons pas ce que chaque trajectoire accomplit. Nous utilisons des techniques de clonage de comportement pour régresser, à chaque pas de temps, l'action de la trajectoire à partir de la caméra du robot. Le point important de cette étape est que cette régression est conditionnée par la représentation de la vidéo constituant la trajectoire complète du robot, obtenue par l'encodeur entraîné précédemment. Ainsi, après cette phase d'entraînement, le robot est capable de reproduire le mouvement de n'importe quelle vidéo de lui même. Cependant, il n'est pas à ce stade directement capable de reproduire des vidéos d'humains.

Déploiement avec des instructions humaines Lors de la phase de déploiement, nous utilisons la mesure de similarité entre vidéos pour chercher, à partir de l'instruction vidéo humaine, une vidéo de robot correspondant à la tâche voulue. En d'autres termes, la mesure de similarité permet au robot de "deviner" le mouvement qu'il doit accomplir. Enfin, nous exécutons la trajectoire en déroulant dans l'environnement robotique à chaque pas de temps la politique neuronale conditionnée par la représentation de la vidéo de robot ainsi obtenue. Ainsi le robot est capable d'effectuer des vidéos humaines qu'il n'a pourtant jamais vues pendant son entraînement.

6.3.3 Résultats expérimentaux

Nous évaluons la méthode proposée sur des environnements de manipulation robotique proposant plusieurs tâches possibles et qui permettent de valider si le robot effectue bien la bonne tâche qui lui est demandée. Ces environnements sont illustrés en Figure 6.5. Dans les environnements "table de travail", la politique observe une image de l'environnement,

contrôle la vélocité du robot et est testée sur les tâches "fermer le tiroir", "pousser la tasse" et "tourner le robinet vers la droite". Les trajectoires de robotique sont collectées en générant des mouvements aléatoires dans l'environnement. Dans les environnements "cuisine", la politique observe une image de l'environnement ainsi que des informations issues des capteurs proprioceptifs du robots, contrôle l'actionnement à effectuer sur chaque joint du robot et est testée sur les tâches "ouvrir le micro-onde", "ouvrir la porte de gauche" et "ouvrir la porte coulissante". Les trajectoires de robotique sont collectées par des politiques expertes pour chaque tâche, mais sans indiquer à quelle tâche chaque trajectoire correspond effectivement. L'enjeux y est ainsi plus focalisé sur la reproduction des trajectoires de manipulations complexes et sur l'utilisation de la similarité pour correctement deviner la tâche effectuée dans la vidéo humaine.

Nos résultats expérimentaux montrent que nos politiques accomplissent la tâche correspondant à l'instruction vidéo la majorité du temps, bien que les vidéos humaines utilisées n'aient pas été vues pendant la phase d'entraînement. Par ailleurs, notre approche obtient des taux de succès plus élevés que l'approche antérieure *DVD* tout en étant bien moins gourmande en ressources de calcul pendant les phases d'entraînements et de déploiement.

6.4 Conclusion

La thèse souligne l'importance cruciale de la rareté des données d'entraînement dans le domaine de l'apprentissage robotique, entravant les progrès vers des robots généralistes. Ainsi, des avancées supplémentaires en apprentissage par renforcement en ligne et en exploitation des données humaines pour l'apprentissage robotique sont nécessaires. Dans ce contexte, cette thèse propose un algorithme d'apprentissage par renforcement plus efficace ainsi qu'un algorithme d'apprentissage de politiques à partir de vidéos de démonstrations humaines. Elle ouvre la voie à des perspectives de recherche nombreuses dans le domaine de l'apprentissage robotique. Par exemple, il pourrait être intéressant de pouvoir doter les robots de la capacité de suivre des instructions humaines exprimées dans n'importe quelle modalité: sous la forme d'images naturelles, de vidéos, d'instructions en langage naturel, ou sous la forme de n'importe quelle combinaison de celles-ci. Par ailleurs, un axe prometteur de recherche est de pouvoir pré-entraîner les politiques neuronales de contrôle robotique directement à partir de vaste jeux de données de vidéos humaines.

RÉSUMÉ

Le développement de robots généralistes capables d'accomplir une vaste gamme de tâches présente un énorme potentiel pour alléger la charge de travail humain dans des tâches physiquement exigeantes, dangereuses ou fastidieuses. Cependant, les progrès de l'apprentissage robotique ont été relativement lents par rapport à d'autres domaines de l'apprentissage automatique, en partie en raison du manque de jeux de données de grande envergure pour la robotique. Cette thèse vise à présenter de nouvelles méthodes pour l'apprentissage des politiques multi-tâches pour la robotique. Dans notre première contribution, nous présentons un nouvel algorithme d'apprentissage par renforcement qui apprend des politiques d'atteinte d'objectifs en interagissant avec l'environnement. Notre approche intègre des sous-objectifs imaginés pour guider l'apprentissage de la politique lors de l'entraînement, ce qui se traduit par une meilleure efficacité d'échantillonnage et la capacité à résoudre des tâches temporellement plus complexes. Dans notre deuxième contribution, nous proposons une méthode pour apprendre des politiques capables de suivre des instructions vidéo humaines dans des environnements de manipulation multi-tâches basés sur la vision. En utilisant un ensemble de données volumineux existant de vidéos humaines annotées, nous parvenons à cela sans avoir besoin de démonstrations robotiques annotées ni de conception de fonctions de récompenses spécifiques pour chaque tâche.

MOTS CLÉS

apprentissage profond, apprentissage par renforcement, vision par ordinateur, robotique

ABSTRACT

Developing versatile robots capable of performing diverse tasks has the potential to alleviate human labor in physically demanding, dangerous, and tedious activities. However, the progress of robot learning has been relatively slow compared to other domains of machine learning partially due to the lack of large-scale robotics datasets. This thesis aims to introduce novel methods for learning multi-task policies for robotics. In our first contribution, we present a novel reinforcement learning algorithm that learns goal-reaching policies by interacting with the environment. Our approach incorporates imagined subgoals to guide policy learning during training, resulting in higher sample efficiency and the ability to solve more complex temporally extended tasks. In our second contribution, we propose a method for learning policies in multi-task vision-based manipulation environments that can follow human video instructions. By utilizing an existing large dataset of labeled human videos, we achieve this without requiring annotated robot demonstrations or task-specific reward shaping.

KEYWORDS

deep learning, reinforcement learning, computer vision, robotics