From Reinforcement Learning to Robot Learning: Leveraging Prior Data and Shared Evaluation

Victoria Dean

CMU-RI-TR-23-36

July 2023

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee

Abhinav Gupta	Carnegie Mellon University (*chair*)
David Held	Carnegie Mellon University
Shubham Tulsiani	Carnegie Mellon University
Rob Fergus	New York University
Chelsea Finn	Stanford Univeristy

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics

Abstract

Deep learning holds promise for learning complex patterns from data, which is especially useful when the input or output space is large. In robot learning, both the input (images or other sensor data) and the output (actions such as joint angles) can be large, suggesting that deep learning could be especially well-suited to making progress on challenging problems in robotics.

However, unlike most machine learning applications, robotics involves physical constraints that make off-the-shelf learning challenging. Robots are expensive and typically require human involvement for resetting environments and fixing hardware. These constraints make large-scale data collection and training difficult, presenting a major roadblock to applying today's data-intensive algorithms. Robot learning has an additional roadblock in evaluation: every physical space is different, making results across labs inconsistent.

Two common assumptions of the robot learning paradigm limit data efficiency. First, an agent typically assumes isolated environments and no prior knowledge or experience – learning is done *tabula-rasa*. Second, agents typically receive only image observations as input, relying on vision alone to learn tasks. However, in the real world, humans learn with many senses across many environments and come with prior experiences as they learn new tasks. This approach is not only practical but also crucial for feasibility in real robotics where it is cost-prohibitive to collect many samples from deployed physical systems.

In this thesis, I present work that lifts these two assumptions, improving the data efficiency of robot learning by leveraging multimodality and pretraining. First, I show how multimodal sensing like sight and sound can provide rich self-supervision (Chapter 2). Second, I introduce a framework for pretraining and evaluating self-supervised exploration via environment transfer (Chapter 3). In Chapter 4, I apply these ideas to real-world manipulation, combining the benefits of large-scale pretraining and multimodality through audio-visual pretraining for contact microphones. Finally, drawing upon the benchmarking efforts from Chapter 3, I introduce a real-robot benchmark for evaluating the generalization of both visual and policy learning methods via shared data and hardware (Chapter 5).

Acknowledgments

First I would like to thank my advisor, Abhinav Gupta. I have enjoyed learning how to give engaging talks and write paper introductions from your example and feedback. From working with undergraduate students to teaching a course to pursuing the robotics cloud concept, my PhD has not followed a linear progression of research. I am grateful to you for recognizing my interests and allowing me to pursue these opportunities, even when these opportunities did not necessarily get me closer to a thesis. Finally, from my very first year, you have talked with me about my goals and helped me strategize. From fellowship applications to internship timing to recommendation letters, thank you for supporting my career ambitions.

I am deeply grateful to my thesis committee members for their wisdom and support. David Held, you taught me how to critique papers in Deep Reinforcement Learning for Robotics, and a year later he taught me about course design when I TAed for the course. Shubham Tulsiani, your support has spanned low-level (how to use the Python Debugger) to high-level (how to give a job talk). Rob Fergus, you have shown me kindness since we met at my first NeurIPS as an undergraduate, and your work's diverse machine learning applications is an inspiration to my own research. Chelsea Finn, your guidance over the past 10 years has shaped my career and my work—from deciding to pursue a PhD to determining what type of robots to buy.

This thesis would not have been possible without my fantastic collaborators. Simone Parisi, with whom I worked on *Interesting Object, Curious Agent*, showed me what thorough experiments look like. Gaoyue Zhou's dedication and real robot expertise made the *Train Offline, Test Online* benchmark come to life. Jared Mejia led the Hearing Touch project with determination and an infectious optimism about research. Thanks also to Mohan Kumar Srirama, Lerrel Pinto, and Yonadav Shavit for supporting the robotics cloud effort and to Tess Hellebrakers for contributing to Hearing Touch. Thanks to my labmates for the shared community and shared infrastructure, especially to Sam Powers, Helen Jiang, Raunaq Bhirangi, and Sudeep Dasari for discussions and support throughout the years. I am also delighted to have interned with Doina Precup at DeepMind Montreal; I left all of our one-on-ones excited about ideas and appreciative of your reinforcement learning wisdom.

Thanks to the undergraduate researchers I've worked with not only for your research contributions, but also for teaching me how to be a better mentor: Shaden Alshammari, Jacob Adkins, Eliot Xing, Krishna Patel, and Maxine Lui. I am grateful to Rachel Burcin and the RISS program for the opportunity to work with amazing summer interns.

For encouraging my teaching career, I am beyond grateful to my Computer Science Pedagogy instructors: Michael Hilton, Charlie Garrod, and Franceska Xhakaj. Without you, I may never have found this special flavor of academia I have grown to love. Thank you Illah Nourbakhsh for further supporting my teaching career in

allowing me to co-instruct Ethics and Robotics. Creating our virtual classroom was one of the most joyful experiences of my PhD, and I learned so much from seeing how you lead discussions and interact with students. I thank the Eberly Center and Jacqui Stimson for the pedagogical training and feedback. I appreciate the friendliness of the CS teaching community and everyone who guided me through the job market this past year.

I am grateful to Matt Johnson-Roberson for listening to student perspectives, making our first-ever PhD Student Retreat possible, and making our department a better place. Thank you Henny Admoni, Zeynep Temel, and the rest of the RI Knitting Club for the chatting and crafting; I always looked forward to our sessions.

I'm grateful to my PhD cohort and the other friends who supported me and kept me having fun. In particular, thanks to Pallavi Koppol, Bailey Flanigan, Tabitha Edith Lee, Kiyn Chin, Allie Del Giorno, Divya Shanmugam, and Esther Brown for everything from poster pitch practice to D&D sessions. Thanks also to the many wonderful housemates I had—especially during the pandemic—for the socializing, meals, and adventures. To my therapist, thank you for the structure and reflection that kept me moving forward when research felt stuck.

Ben, you've been the best spouse I could ask for. Thank you for always being there to listen, read a paper draft, or go on a walk. To my parents, thank you for raising me to be curious and supporting my academic pursuits, even when it meant late nights chaperoning high school students in a robotics lab.

Lastly, I appreciate the numerous people not listed here whose support has gotten me to this point: mentors, teachers, colleagues, and family. Thank you for helping me grow—I hope to make you proud.

Contents

Chapter 1

Introduction

Imagine a baby playing with an object she has never seen before. She shakes it and listens to the noise. She watches her parent to see how it is used. While these behaviors might not seem intelligent to adults, babies learn in ways that exploit the richness of their early lives. They make use of all their senses; they seek creative forms of feedback; they thrive on the diversity of the world around them [133]. I see these behaviors as inspiration for building better agents. Namely, I aim to improve robot learning by leveraging self-supervision, multimodality, and prior experience.

Reinforcement learning (RL) allows systems to move beyond passive learning and interact with the world while learning from these interactions. In the standard RL paradigm, a researcher manually specifies a reward function (score), and an agent learns to maximize this reward. This works well in games like Atari or Go, but in applications like robotics, reward functions are hard to formulate and learning from real-world data requires sample efficiency. The challenges in RL can be grouped into two areas: how to collect interesting data in an environment (exploration) and how to learn tasks from such data (policy learning). In my thesis, I explore how to improve both exploration and policy learning to make RL feasible in real-world settings.

Current frameworks for RL exploration are poor proxies for the ways children explore the world. RL agents tend to begin tabula-rasa (initialized from scratch in a single environment) and use only vision or state vectors, missing out on other sensory modalities. In this thesis, I aim to put exploration more in line with the real world: an agent uses large-scale data (from prior environments and passive sources) to effectively transfer knowledge to a new setting, where self-supervision and multimodality guide quick adaptation.

1.1 Overview

The first prong of this thesis (Chapter 2) is about using additional sensory modalities. Multimodal approaches are especially amenable to self-supervised methods

because we can use two complementary modalities as joint supervision. In See Hear Explore [41], we showed how an agent perceiving multiple modalities (like vision and sound) can learn associations between them as an informative reward signal.

The second prong (Chapter 3) shows how policy transfer can accelerate training. We started down this route by designing an evaluation protocol: an agent first learns to explore across many environments without a specified goal and then explores new environments to solve new tasks [117]. In this setting, our proposed approach improved sample efficiency through more expressive self-supervision that rewarded novel environment changes, and the evaluation protocol brought richness through pretraining in multiple environments.

These first two prongs demonstrate how policy learning can move beyond tabula-rasa, state-based exploration to something more suitable for the real world. In the second half of this thesis, I bring these ideas to real robotic manipulation. In Chapter 4, I combine multimodality with large-scale pretraining to improve the performance of real-world manipulation, mitigating the amount of costly trial-and-error data required. We use contact microphones as an alternative tactile sensor that captures inherently audio-based information, enabling large-scale audio-visual pretraining for robotics.

The final aspect of my thesis provides a way to evaluate robot learning from diverse, passive data. Following the robotics cloud concept [42], Chapter 5 introduces the Train Offline, Test Online (TOTO) benchmark, a shared setting for conducting experiments on real, remotely-operable robots. This benchmark allows new researchers to contribute to the field without purchasing hardware, re-implementing baselines, or collecting their own data, enabling better comparison of more approaches. While the full scale of this vision is not in scope for this thesis, I show its feasibility and desirability through the TOTO benchmark, which evaluates the generalization of visual representations and policies in a shared manipulation environment.

The multimodality and pretraining aspects of my work move us towards data-efficient real robot learning, while the Robotics Cloud makes the deployment of such methods possible on a broader scale.

In Chapter 4, I will combine multimodality with large-scale pretraining to improve real-world manipulation, mitigating the amount of costly trial-and-error data required. We use contact microphones as an alternative tactile sensor that captures inherently audio-based information, allowing us to leverage large-scale audio-visual pretraining.

The final aspect of my thesis provides a way to evaluate robot learning from diverse, passive data. Chapter 5 motivates the Robotics Cloud [42], a shared setting for conducting experiments on real, remotely-operable robots. This cloud would allow new researchers to contribute to the field without purchasing hardware, re-implementing baselines, or collecting their own data, enabling better comparison of more approaches. While the full scale of this vision is not in scope for this thesis,

I show its feasibility and desirability through the Train Offline, Test Online (TOTO) benchmark. In particular, TOTO evaluates the generalization of visual representations and policies in a shared manipulation environment.

The multimodality, self-supervision, and data diversity aspects of my work move us towards tractable real robot learning, while the Robotics Cloud makes the deployment of such methods possible on a broader scale.

Chapter 2

See, Hear, Explore: Curiosity via Audio-Visual Association

2.1 Introduction

Many successes in reinforcement learning (RL) have come from agents maximizing a provided extrinsic reward such as a game score. However, in real-world settings, reward functions are hard to formulate and require significant human engineering. On the other hand, humans explore the world driven by intrinsic motivation, such as curiosity, often in the absence of rewards. But what is curiosity and how would one formulate it?

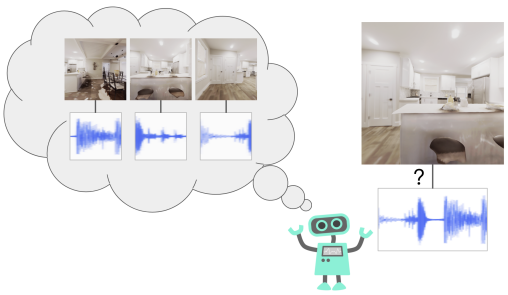


Figure 2.1: **See, Hear, Explore**: We propose a formulation of curiosity that encourages the agent to explore novel associations between modalities, such as audio and vision. In Habitat, shown above, our method allows for more efficient exploration than baselines.

Recent work in RL [20, 119, 121] has focused on curiosity using future prediction. In this formulation, an exploration policy receives rewards for actions that lead to differences between the real future and the future predicted by a forward dynamics model. In turn, the dynamics model improves as it learns from novel states. While the core idea behind this curiosity formulation is simple, putting it into practice is quite challenging. Learning and modeling forward dynamics is still an open research problem; it is unclear how to handle multiple possible futures, whether to explicitly incorporate physics, or even what the right prediction space is (pixel space or some latent space).

The use of multiple modalities in *human* learning has a long history. Research in psychology has suggested that humans look for incongruity [76]. A baby might hit an object to hear what it sounds like. Have you ever found yourself curious to touch a material different from anything you have seen before? Humans are drawn towards discovering and exploring novel associations between different modalities. Dember and Earl [44] argued that intrinsic motivation arises with discrepancy between expected sensory perception and the actual stimulus. More recent work has shown the presence of multimodal stimulation and exploration in infants [57, 168]. In cognitive development, both sight and sound guide exploration: babies are drawn towards colorful toys that squeak and rattle [105].

Inspired by human exploration, we introduce See Hear Explore (SHE): a curiosity for novel associations between sensory modalities (Figure 2.1). SHE rewards actions that generate novel associations (shared information) between different sensory modalities (in our case, pixels and sounds). We first demonstrate that our formulation is useful in several Atari games: SHE allows for more exploration, is more sample-efficient, and is more robust to noise compared to existing curiosity baselines on these environments. Finally, we show experiments on area exploration in the realistic Habitat simulator [138]. Our results demonstrate that in this setting our approach significantly outperforms baselines.

To summarize, our contributions in this work include: 1) SHE, a curiosity formulation that searches for novel associations in the world. To the best of our knowledge, multimodal associations have not been investigated in self-supervised exploration; 2) we show our approach outperforms the commonly-used curiosity approaches on standard Atari benchmark tasks; 3) most importantly, multimodality is one of the most basic facets of our rich physical world (audio and vision are generated by the same physical processes [164]). We show experiments on realistic area exploration in which SHE significantly outperforms baselines. This work builds on efficient exploration, which will be crucial as we push agents to explore more complex unknown environments.

2.2 Related Work

Our work uses audio as an additional modality for self-supervised exploration. We divide the prior work into two categories: exploration (Section 2.2.1) and multimodal learning (Section 2.2.2).

2.2.1 Exploration

Prior exploration work has used error [1, 119, 140, 150], uncertainty [75, 121, 152], and potential improvement [139] of a prediction model as intrinsic motivation. Some methods use count- or pseudo-count-based exploration [12, 158]. Others use auxiliary losses to supplement rewards and improve sample efficiency [77, 148].

One popular approach is curiosity by self-supervised prediction [20, 119]. In this form of curiosity, an intrinsic reward encourages an agent to explore situations with high error under a jointly-trained future prediction model. The model's error is a proxy for novelty: unpredictable situations are more likely novel and therefore ones the agent should explore. These future-predicting models can be difficult to train, especially in visual space. Our method also looks at self-supervised exploration, but our intrinsic reward does not rely on future prediction. We circumvent the need for future prediction by leveraging multimodality. SHE rewards association classification error (i.e. association novelty) as opposed to higher-dimensional prediction error. Our key insight is that *associative* models across modalities are simpler to learn, and their accuracy is also indicative of novelty.

2.2.2 Multimodal Learning

Multimodal settings are especially amenable to self-supervision, as information from one modality can be used to supervise learning for another modality. One prior work learned a joint visual and language representation using Flickr images and associated descriptors [149]. In computer vision, audio can provide additional information that complements images [8, 61, 115]. Recent work [27, 59] has looked at audio-visual embodied navigation, in which audio is emitted from a goal point to aid in supervised learning of navigation. In the same environment, Gao et al. [62] used audio and visual information for learning visual feature representations. We test on this audio-visual navigation environment, but for unsupervised exploration in RL; we have no goal states.

Audio and visual information are closely linked, and since we commonly have access to both in the form of video, this is a rich area for self-supervision. Aytar et al. [9] used audio from Atari in the form of YouTube videos of people playing the games. This work uses audio-visual demonstrations from YouTube to learn a visual embedding. The setup here is learning from demonstrations from humans. In our case, on the other hand, the audio-visual associations drive intrinsically motivated exploration. We learn multimodal alignment from active data, which the agent both collects and uses.

In robotics settings, the use of additional modalities such as tactile sensing [24, 107] or audio [34] is increasingly popular for grasping and manipulation tasks. Lee et al. [94] showed the effectiveness of self-supervised training of tactile and visual representations by demonstrating its use on a peg insertion task. While these previous approaches have demonstrated the benefits of using multiple sensory modalities for learning better representations or accurately solving tasks, in this work we demonstrate its utility for allowing agents to explore. To the best of our knowledge, using audio to learn actions for exploration is unique to our work.

2.3 See, Hear, Explore

We now describe SHE, our exploration method based on associating audio and visual information. Our goal is to develop a form of curiosity that exploits the multimodal nature of the input data. Our core idea is that the SHE agent learns a model that captures associations between two modalities. We use this model to reward actions that lead to unseen associations between the modalities. By rewarding such actions, we guide the exploration policy towards discovering new combinations of sight and sound.

More formally, we consider an agent interacting with an environment that contains visual and sound features, which we call $x_t = (v_t, s_t)$ for time t where v_t is the visual feature vector and s_t is the sound feature vector. The agent explores using a policy $a_t \sim \pi(v_t; \theta)$ where a_t corresponds to an action taken by the agent at time t. To make for easier comparison to visual-only baselines, our agent is only given access to the visual features v_t and not the audio features s_t. To enable this agent to explore, we train a discriminator D that tries to determine whether an observed multimodal pair (v_t, s_t) is novel, and we reward the agent in states where the discriminator is surprised by the observed multimodal association.

2.3.1 Why Novel Associations?

The goal of an exploration policy is to perform actions that uncover states that lead to a better understanding of the world. One commonly used exploration strategy involves rewarding actions that lead to unseen or novel states [12]. While this strategy seems intuitive, it does not handle the fact that while some states might not have been seen, we still understand them and hence they do not need to be explored. In light of this, recent approaches have used a prediction-based formulation. If a model cannot predict the future, it needs more data points to learn. However, sometimes we may have seen enough examples, and prediction is still challenging, leading a prediction-based exploration policy to get stuck. For example, consider the couch-potato issue: the random TV in the Unity environment (as described in Burda et al. [20]) yields high error for prediction models, so prediction-based curious agents receive high rewards for staring at the TV, though this is not a desirable type of exploration.

Trying to avoid these problems has shaped much of the work on intrinsic motivation; Schmidhuber [139], Oudeyer et al. [114], White et al. [167], and Burda et al. [21] all formulate intrinsic rewards with the goal of mitigating problems like the couch-potato agent. Our approach, different from this body of prior work, looks at how multimodal data can mitigate these issues.

Our underlying hypothesis is that discovering new sight and sound associations will help mitigate the shortcomings of the previously described count-based and prediction-based exploration strategies. By using an association model, we ask a simpler question: can this image co-occur with this sound? Consider another ex-

ample in which pressing a button randomly produces one of 3 distinct sounds. Our approach could learn to classify all as associated, while an agent using future prediction error would always be curious. This focus on association effectively helps ignore stochasticity, mitigating the couch-potato problem by focusing on non-random structure. Such a model can allow generalization to unseen states and also does not need to predict the future to provide an informative signal for exploration.

2.3.2 Association Novelty via Alignment

The core of our method is the ability to determine whether a given pair (v_t, s_t) represents a novel association. To tackle this problem, we learn a model in an online manner. Given past trajectories, a model learns whether a certain audio-visual input comes from a seen or new phenomenon. One way to model this would be to use a generative model such as a VAE [82] or GAN [65], which could determine if the image-audio combination is within the distribution or out of distribution. However, generative models are also difficult to train, so we propose using a discriminator to predict if the image-audio pair is novel, which has a smaller, binary output space.

We train this discriminator to distinguish real audio-visual pairs from 'fake' pairs from another distribution, with the insight that the learned model is more likely to classify novel pairs as fake. Here, the observed image-audio pairs during exploration act as positive training examples, but a critical question is how to obtain negative image-audio pairs. To this end, we reformulate the problem as whether image-audio pairs are aligned or not: we obtain 'fake' samples by randomly misaligning the audio and visual modalities, similar to Owens and Efros [115]. The positive data is then the aligned image-audio pairs, and the negative data is comprised of misaligned ones. The discriminator model, as shown in Figure 2.2, outputs values between 0 and 1, with 1 representing high probability of audio-visual alignment and 0 representing misalignment. We can then leverage the misalignment likelihood as an indicator of novelty since the discriminator would be uncertain in such instances.

2.3.3 Training

Having introduced association novelty via alignment, we now describe how we implement this idea using function approximators. During training, the agent policy is rolled out in parallel environments. These yield trajectories which are each chunked into 128 time steps. A trajectory consists of pairs of preprocessed visual and sound features: $(v_1, s_1), (v_2, s_2)...(v_{128}, s_{128})$. These trajectories are used for two purposes: 1) updating the discriminator D as described below and 2) updating the exploration policy based on the intrinsic reward r_t^i (computed using the discriminator), also described below.

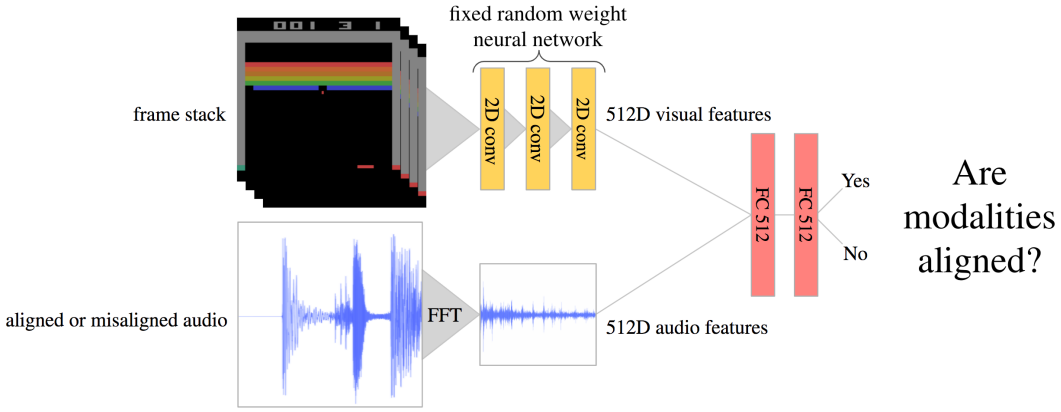


Figure 2.2: **Our audio-visual association model**: The frames (top left) and potentially misaligned audio waveform (bottom left) are preprocessed into 512-dimensional feature vectors using a random feature network and FFT, respectively. The discriminator network (right) takes these features as inputs and is trained to output whether or not they are aligned. 2D conv represents a standard convolutional layer and FC 512 represents a fully-connected layer with 512 units.

Training the Alignment Discriminator The discriminator D is a neural network that takes a visual and sound feature pair as input and outputs an alignment probability. To train D, we start with positive examples from the visual and sound feature pairs (v_t, s_t). With 0.5 probability we use the true aligned pair, and with 0.5 probability we create a false pair consisting of the true visual feature vector v_t and a sound feature vector uniformly sampled from the current trajectory. We call this false sound s'_t. We define a binary variable z_t to indicate whether the true audio was used, i.e. when we give the discriminator the true audio s_t, we set $z_t = 1$, and when we give the discriminator the false audio s'_t, $z_t = 0$. We use a cross-entropy loss to train the discriminator, similar to prior work [9, 115]:

$$\mathcal{L}_t(v_t, s_t, s'_t, z_t) = \begin{cases} -\log(D(v_t, s_t)), & \text{if } z_t = 1 \\ -\dfrac{||s_t - s'_t||_2}{\mathbb{E}_{\text{batch}}||s_t - s'_t||_2} \log(1 - D(v_t, s'_t)), & \text{if } z_t = 0 \end{cases}$$

In the $z_t = 0$ case above, we weight the cross-entropy loss to prevent the discriminator from being penalized in cases where the true and false audio are similar. We weight by the L2 difference between the true and false audio feature vectors and normalize by dividing by the mean difference across samples in the batch of 128 trajectories. This loss is used for updating the discriminator and is not used in computing the agent's intrinsic reward.

Training the Agent via Intrinsic Reward We want to reward actions that lead to unseen image-audio pairs. For a given image-audio pair, if the discriminator predicts 0 (unseen or unaligned), we want to reward the agent. On the other hand, if the discriminator correctly outputs 1 on a true pair, the agent receives no reward. Mathematically, the agent's intrinsic reward is the negative log-likelihood of the discriminator evaluated on the true pairs: $r_t^i := -\log(D(v_t, s_t))$, where the output of D is between 0 and 1. Audio-visual pairs that the discriminator knows to be aligned get a reward of 0, but if the discriminator is uncertain (the association surprised the discriminator) the agent receives a positive reward. The agent takes an action and receives a new observation v_t and intrinsic reward r_t^i (note that the agent does not have access to the sound s_t). The agent is trained using PPO [142] to maximize the expected reward: $\max_\theta \mathbb{E}_{\pi(v_t;\theta)}\left[\sum_t \gamma^t r_t^i\right]$. *The agent does not have access to the extrinsic reward. Extrinsic reward is used only for evaluation.* This will enable the use of our method on future tasks for which we cannot easily obtain a reward function. See Burda et al. [20] for further discussion on training with no extrinsic reward while using it for evaluation.

2.4 Experiments

In this section, we will test our method in two exploration settings (Atari and Habitat) and compare it with commonly-used curiosity formulations.

2.4.1 Environments

Atari Similar to prior work, we demonstrate the effectiveness of our approach on 12 Atari games. We chose a subset of the Atari games to represent environments used in prior work and a range of difficulty levels. We excluded some games due to lack of audio (e.g. Amidar, Pong) or the presence of background music (e.g. RoadRunner, Super Mario Bros). The action space is different from the one used in the future prediction curiosity work [20], as we use Gym Retro [110] in order to access game audio, and Retro environments use a larger action space. The original work reported results using the minimal action space, Discrete(4), whereas we use Discrete(6). We note that the larger action space does slow exploration, but it is used for both our method and the baselines for fair comparison. To compute audio features, we take an audio clip spanning 4 time steps (1/15th of a second for these 60 frame per second environments), apply a Fast Fourier Transform, and downsample using max pooling to a 512-dimensional feature vector. This vector is used as input to the discriminator along with a 512-dimensional visual feature vector.

Habitat Navigation We also test our method in a navigation setting using Habitat [138] (Figure 2.3). In this environment, the agent moves around a photorealistic Replica scene [153]. We use the largest Replica scene, Apartment 0, which has

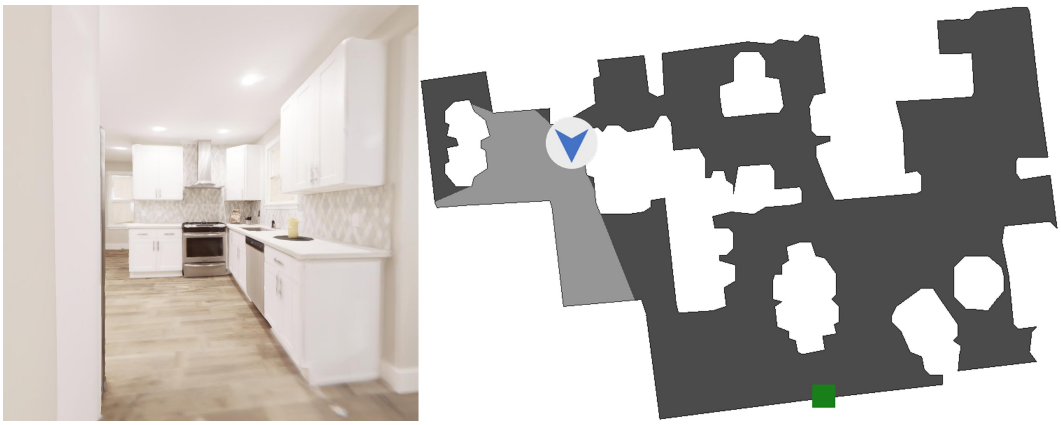


Figure 2.3: **Habitat visualization**: Left: an example agent view. Right: the top-down map for apartment 0 (not seen by agent). The agent is the blue arrow and the audio source is the green square. Gray areas are open space, while white areas are obstacles, which make exploration challenging.

211 discrete locations. In each location, the agent can face in 4 directions. At each timestep, the agent takes one of 3 discrete actions: turn left, turn right, or move forward. As in our Atari experiments, the agent is not given any extrinsic reward; we simply want to see how well it can explore the area without supervision. We use the audio-visual navigation extension from Chen et al. [27], which emits a fixed audio clip from a fixed location and allows our agent to hear the sound after simulating room acoustics. The perceived sound at each time step is less than 1 second long, and we zero pad this audio to 1 second to make each sound equal length for feature computation. We apply FFT and downsample to a 512-dimensional feature vector, the same as done in Atari, described above.

2.4.2 Baselines

We compare to future prediction curiosity [20], which as previously described performs visual future prediction. We build upon the open-source code from the authors (see the appendix for more details). We also compare to exploration via disagreement [121] and Random Network Distillation (RND) [21]. We use the same hyperparameters (which were optimized for the future prediction and disagreement baselines) for policy learning across all approaches. We use random CNN features [20, 21] for the visual feature representation for our method and the baselines in all experiments.

2.4.3 Atari Experimental Results

We trained our approach and baselines for 200 million frames using the *intrinsic* reward and measure performance by the *extrinsic* reward throughout learning. Figure 2.4 shows these results. Each method was run with three random seeds, and the plots show the mean and standard error for each method. Please see the appendix for more experimental details. Across many environments, our method enables better exploration (as judged by the extrinsic reward) and is more sample efficient than the baselines. Of the 12 environments, SHE outperforms the disagreement baseline in 9 and the future prediction baseline in 8. We hypothesize that states leading to novel audio-visual associations, such as a new sound when killing an enemy, are more indicative of a significant event than ones inducing high prediction error (which can happen due to inaccurate modeling or stochasticity) and this is why our approach is more efficient across these environments.

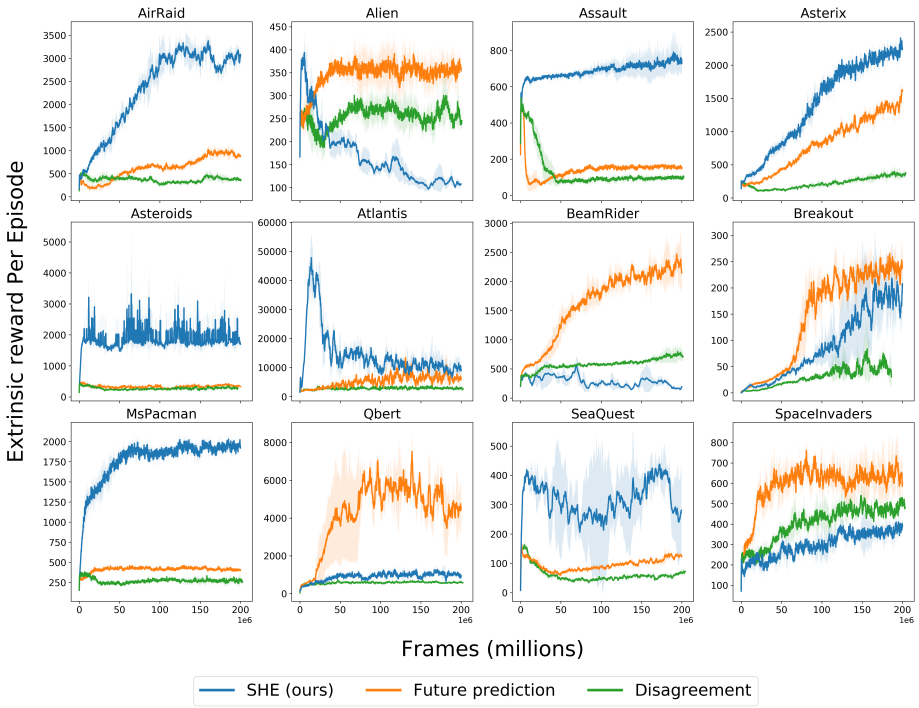


Figure 2.4: **Atari training curves**: Average extrinsic reward (never seen by the agent) throughout training for our method, future prediction [20], and exploration via disagreement [121]. Our method outperforms the baselines in 8 of 12 environments, supporting our hypothesis that audio-visual association is a useful signal for accelerating exploration.

Understanding Failure Cases While our approach generally exceeds the performance of or is comparable to the baselines, there are some environments where SHE underperforms. We have analyzed these games and found common failure cases: 1) Audio-visual association is trivial. For example in Qbert, the discriminator easily learns the associations: every time the Qbert agent jumps to any cube the same sound is made, thus making the discriminator's job easy, leading to a low agent reward. Visiting states with already learned audio-visual pairs is necessary for achieving a high score, even though they may not be crucial for exploration. The game Atlantis had similarly high discriminator performance and low agent rewards. 2) The game has repetitive background sounds. Games like SpaceInvaders and BeamRider have background sounds at a fixed time interval, but these sounds are hard to visually associate. Here the discriminator has trouble learning basic cases, so the agent is unmotivated to further explore. In Alien, the agent quickly learns that by quickly passing from one side of the screen to the other, a sound occurs with a slight delay that makes it hard to align with the frame. The agent learns to repeat this trick continuously, putting the discriminator in a situation like 2).

Hard Exploration Environment According to Taïga et al. [157], Gravitar is a hard exploration environment. Such environments are particularly difficult to solve without learning from demonstrations [9], using extrinsic reward [157], or exploiting structure in the game [52]. Even for humans, it can be unclear how to play Gravitar upon first introduction, in contrast with other Atari games that are intuitively simple. Despite Gravitar's difficulty, SHE allows the agent to explore well, while the baselines perform poorly (Figure 2.5). After examining the game, we hypothesize that the game's visual dynamics are not that interesting on their own, but the audio-visual associations are. We also applied our method and the baselines to other hard exploration games, but in these cases, no method was successful in the training time allotted.

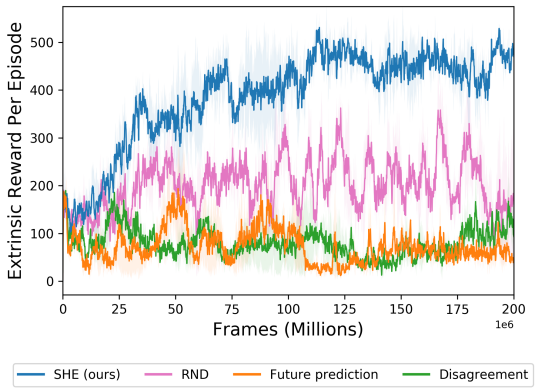
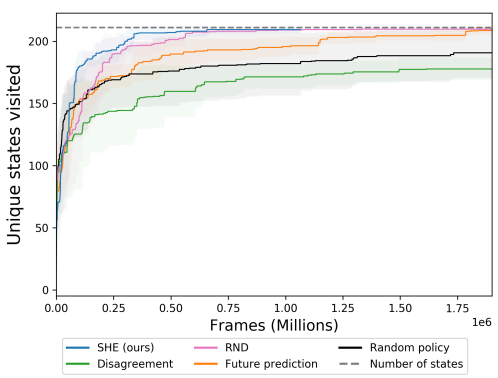
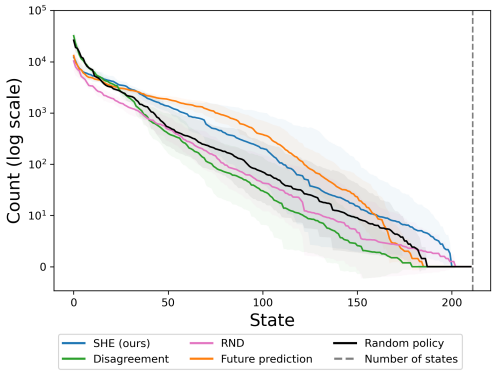


Figure 2.5: **Case study on Gravitar**: Our method is able to explore this hard environment, while baselines have negligible increase in extrinsic rewards.

2.4.4 Habitat Experimental Results



(a) State coverage: unique states visited throughout training. Our method achieves full state coverage about 3 times faster than future prediction curiosity.



(b) State counts: the number of times each state is visited in the first 2000 episodes, sorted by frequency and shown on a log scale. Our method has a wider tail, visiting rare states one to two orders of magnitude more frequently than the baselines.

Figure 2.6: **Habitat exploration results for SHE and baselines**. Each method is run with three different seeds and each seed uses a different start location.

Here we present results from unsupervised area exploration in the biggest scene in Replica [153] with realistic acoustic responses [27]. Figure 2.6 shows the quantitative results. SHE (blue) has similar coverage to RND and reaches full state coverage 3 times faster than future prediction curiosity (Figure 2.6a). We can also look at how much each state is visited (Figure 2.6b). A good exploration method will have higher counts in the rare states. Our method visits these rare states (Figure 2.6b right) about 8 times more frequently than the next-best baseline. It does so by visiting common states (Figure 2.6b left) less frequently. SHE's strong performance on this more realistic task holds promise for future work exploring the real world.

2.4.5 Ablations

Audio in baseline One hypothesis for why our method outperforms baselines is that SHE has access to additional information in the form of audio. To test the benefit of including audio without the use of our association method, we created two additional baselines: an audio-visual prediction baseline and an audio-visual random network distillation baseline. In the audio-visual prediction baseline, the prediction space is concatenated audio and visual features: the future prediction model takes an audio-visual feature vector as input and predicts an audio-visual feature vector. Similarly, in the audio-visual random network distillation baseline, the audio and visual features are concatenated and used as inputs to both the random target network and the predictor network. As the results in the appendix indicate, this does not lead to significant improvement over the visual-only baselines.

Robustness to noise Predicting the future can be especially difficult in the face of inherent uncertainty. To analyze our approach in such a setting, we created a noisy version of the environments, where Gaussian noise is added to the audio and visual feature vector inputs. Our approach can be affected by noise in both audio and visual observations, whereas the baseline is only affected by the visual noise. For these experiments, we chose three environments: one where our method was better (MsPacman), one where the baseline was better (SpaceInvaders), and one where both methods performed well (Asterix). Figure 2.7 depicts results across these three environments both with and without noise. We observe that future prediction curiosity is not robust to such noise: the performance degrades significantly in both Asterix and SpaceInvaders. In contrast, as our approach only relies on associations, it is more robust to such noise.

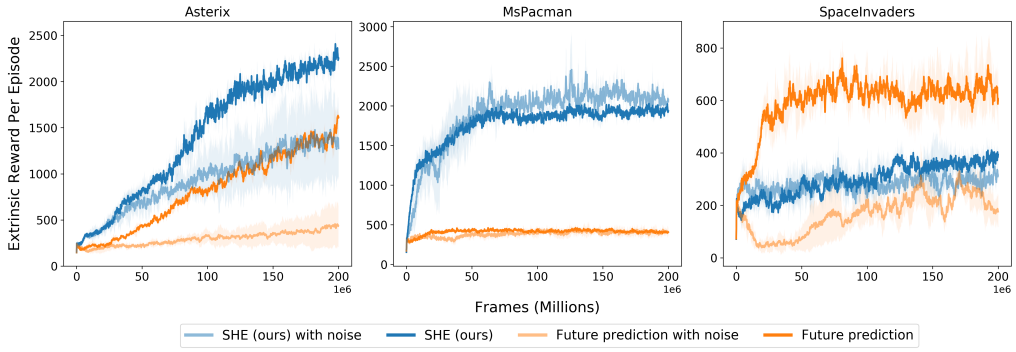


Figure 2.7: **Effect of input noise on performance**: Our method (blue) maintains similar performance with the introduction of noisy observations, while the baseline performance (orange) degrades.

Multiple Curiosity Modules Curiosity can have multiple forms, e.g. prediction-based and multimodal, and these are complementary to each other. To demonstrate this, we ran a joint method combining intrinsic rewards: we sum the losses from future prediction and the audio-visual discriminator. The resulting method is better than the visual-only baseline in 10 of 12 games, sometimes surpassing both (see the appendix for the detailed results).

2.5 Conclusion

Multimodality is one of the most basic facets of our rich physical world. Our formulation of curiosity enables an autonomous agent to efficiently explore a new environment by exploiting relationships between sensory modalities. With results on

Atari games, we demonstrated the benefit of using audio-visual association to compute the intrinsic reward. Our method showed improved exploration over baselines in several environments. The most promise lies in our approach's significant gains when used on a more realistic task, exploration in the Habitat environment, where audio and visual are governed by the same physical processes. We anticipate multimodal agents exploring in the real world and discovering even more interesting associations. Instead of building robots that perform like adults, we should build robots that can learn the way babies do. These robots will be able to explore autonomously in real-world, unstructured environments.

Broader Impact

The lasting impact of RL will be from these algorithms working in the real world. As such, our work is centered around increasing sample efficiency and adaptability. By leveraging self-supervision, we can avoid cumbersome reward shaping, which becomes exponentially more difficult as tasks grow more complex. Although our work here uses simulated agents, our longer-term goal is to deploy multimodal curiosity on physical robots, enabling them to explore in a more sample-efficient manner. Multimodal learning could have a near-immediate impact in autonomous driving, where different sensory modalities are used for perception of near, far, small, and large entities.

Autonomous RL agents have many potential positive outcomes, such as home robots aiding elderly people or those with disabilities. They will save time and money in many sectors of industry. However, they also have the potential to displace parts of the workforce [18].

There could be privacy concerns if merged multimodal data is hard to anonymize or de-identify. There could also be privacy concerns with respect to recording audio data in the wild [92]. With unsupervised RL, it can be hard to predict what behaviors will be learned. For example, a robot using our algorithm might learn to damage sensors to create novel associations. The inability to predict agent behavior can make ensuring safety difficult, which would have consequences in safety-critical settings like autonomous driving or healthcare. Some work has been done on safety in RL [63], and there is more to be done, especially on analyzing the safety of RL exploration policies during training.

Chapter 3

Interesting Object, Curious Agent: Learning Task-Agnostic Exploration

3.1 Introduction

Exploration is one of the key unsolved problems in building intelligent agents capable of behaving like humans. In reinforcement learning (RL), exploration is usually studied under two different settings. The first is task-driven exploration, where the reward is well-defined and the agent's goal is to explore in order to maximize long-term rewards. However, in real life, external rewards are either sparse or unknown altogether. In this setting, exploration is task-agnostic: given a new environment, the agent has to explore it in absence of any external reward. Common approaches to encourage task-agnostic exploration use intrinsically motivated rewards such as prediction curiosity [120, 140], empowerment [130], or visitation counts [12, 113]. But does this setup represent how humans explore?

We argue that the commonly-used task-agnostic exploration setup is unrealistic, both from practical and academic viewpoints. This setup assumes environments in isolation and agents exploring tabula-rasa, i.e., with no prior knowledge or experience. By contrast, we as humans do not learn from one environment in isolation and we do not throw away our past knowledge every time we encounter a new environment [50]. Exploration is rather a lifelong process: every time we encounter new environments, we use our prior knowledge and experience to develop new efficient exploration strategies. We view the exploration problem from a continual learning lens. More specifically, in this setup, the learning agent interacts with one or many environments without any extrinsic goal, *learning to explore* the environments. Later on, the agent effectively transfers the learned *exploration policy* to explore new environments, rather than exploring the new environment tabula-rasa.

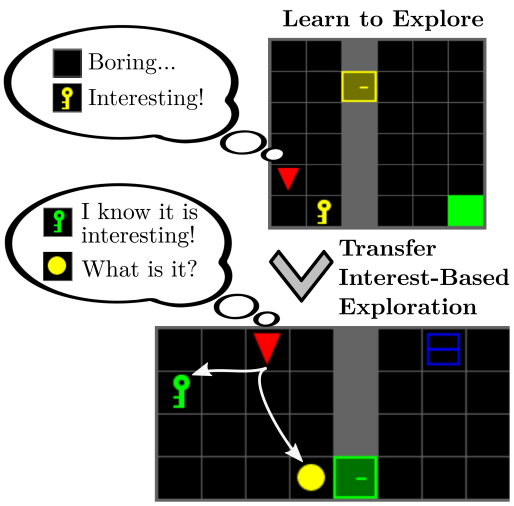


Figure 3.1: **Change-Based Exploration Transfer (C-BET)** trains task-agnostic exploration agents that transfer to new environments. Here the agent learns that keys are interesting, as they allow further interaction with the environment (opening doors). Later, when tasked with reaching a box behind a door, the agent starts by picking up the key.

A key question in learning how to explore is what to learn and how to transfer prior knowledge from one environment to another. Most existing task-agnostic exploration approaches, such as visitation counts, curiosity, or empowerment, define intrinsic rewards in an *agent-centric* manner: they encourage exploration of unseen parts of the environment based on the agent's own belief. In these approaches, exploration is driven by what the agent knows about the world. However, most do not make a distinction between what the agent believes it is interested in and states that would make any agent interested. For example, if the agent uses a visitation count model and has seen many objects of one kind in one environment, it would not explore the same type of objects again in a new environment. This seems to be in stark contrast to how humans explore. Consider a switch with a bell sign. Even though we might have pressed hundreds of doorbell switches (and even this instance), we are still attracted to press it. Some objects in the world just demand curiosity. We argue that apart from an 'agent-centric' component, there is an 'environment-centric' component to exploration, which can be learned from prior knowledge and experiences.

We propose a paradigm change to move away from stand-alone isolated task-agnostic environment exploration to a more realistic multi-environment transfer-exploration setup[1]. We show how to learn exploration policies both from single- and multi-environment interaction, and how to transfer them to unseen environments. This transfer-exploration setup allows agents to use prior experiences for learning task-agnostic exploration. Notably, classic stand-alone task-agnostic approaches were designed for tabula-rasa exploration and hence only explore in an agent-centric manner. They fail to capture the inherent interestingness of some environment components. With this insight, we propose *Change-Based Exploration Transfer* (*C-BET*), a simple yet effective approach learning joint agent-centric and environment-centric exploration. The key idea is for an agent to seek out both sur-

[1]While it can be argued that the real world has no explicit distinction between training and testing, we use this dichotomy only for the purpose of evaluation.

prises (unseen areas) and high-impact (interesting) components of the environment. The experiments show that C-BET (a) learns more effectively when placed in a multi-environment setup, and (b) either outperforms or performs competitively with prior methods across several unseen testing environments. We hope this work will inspire exploration research to focus more on learning from multiple environments and transferring experiences rather than tabula-rasa exploration.

3.2 Preliminaries and Related Work

We consider environments governed by Markov Decision Processes (MDPs), in which an agent observes the state of the environment s and selects actions a according to a policy $\pi(a|s)$. In turn, the environment changes, providing a new observation s' and a reward r. Through environment interaction, the agent collects episodes, i.e., sequences of states, actions and rewards $(s_t, a_t, r_t)_{t=1...T}$. The goal is to learn a policy maximizing the sum of rewards during episodes, i.e., the return.

In this setting, exploration poses many questions. If the environment provides no rewards, what should the agent look for? When should it act greedily with respect to the rewards it has found and stop looking for more? In the history of RL, many approaches have been proposed to tackle these questions. On one hand, classic single-environment approaches range from intrinsic motivation with visitation counts [6, 12, 47, 79, 154], optimism [5, 15, 78, 81, 90], or curiosity [21, 74, 120, 137, 143, 151], to bootstrapping [46, 112] or empowerment [84, 130]. On the other hand, we find approaches to incrementally learn tasks, such as transfer learning [166], continual learning [83], curriculum learning [109], and meta learning [129].

Intrinsic motivation. Exploration strategies relying on intrinsic rewards date back to Schmidhuber [140], who proposed encouraging exploration by visiting hard-to-predict states. More recently, the idea of auxiliary rewards to make up for the lack of external rewards has been extensively studied in RL, supported by evidence from psychology and neuroscience [67]. Several intrinsic rewards have been proposed, ranging from visitation count bonuses [12, 154] to bonuses based on prediction error of some quantity. For example, the agent may learn a dynamics model and try to predict the next state [74, 120, 141, 151]. By giving a bonus proportional to the prediction error, the agent is incentivized to explore unpredictable states. Schultheis et al. [143], instead, proposed to learn intrinsic rewards function by maximizing extrinsic rewards by meta-gradient.

However, in these approaches exploration is agent-centric, i.e., based on agent belief such as the forward model error. In contrast, with this work we propose additionally learning *environment-centric* exploration policies. C-BET neither requires a model nor knowledge of extrinsic rewards. Instead, it encourages the agent to perform actions causing *interesting changes* to the environment. We should note that while Raileanu and Rocktäschel [127] proposed a similar approach, their exploration policy lacks the transfer component and also requires to learn models.

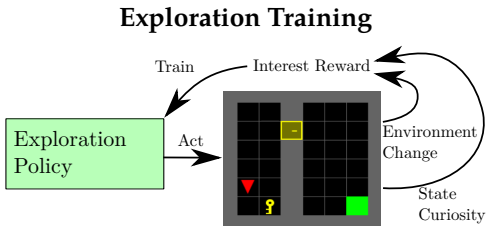


Figure 3.2: **C-BET pre-training.** Our agent interacts with environments and learns using intrinsic rewards computed from state and change counts.

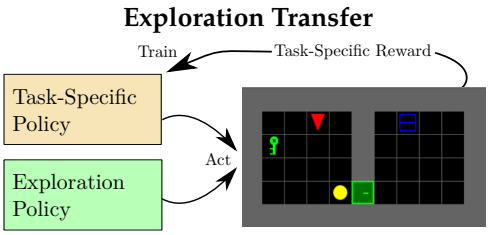


Figure 3.3: **C-BET transfer.** The pre-trained exploration policy is fixed and guides task-specific policy learning in new environments.

Transfer learning. The idea of agents capable of incrementally learning tasks is well-known in the field of machine learning, with the first approaches dating back to the 90s' [131, 132, 161]. In RL, recent methods have focused on policy and feature transfer. In the former, a pre-trained agent (teacher) is used to transfer behaviors to a new agent (student). Examples include policy distillation, where the student is trained to minimize the Kullback-Leibler divergence to the teacher [136] or to multiple teachers at the same time [159]. Alternative approaches, instead, directly reuse policies from source tasks to build the student policy [11, 55, 70]. In feature transfer, a pre-learned state representation is used to encourage exploration when tasks are presented to the agent [71, 169]. Similar to transfer RL, continual RL studies how learning on one or more tasks can help accelerate learning on different tasks, and how to prevent catastrophic forgetting [83, 134, 144]. Meta RL, instead, tries to exploit underlying common structures between tasks to learn new tasks more quickly [56, 129].

However, the setup in these approaches is not task-agnostic, i.e., task-specific policies are transferred rather than exploration policies. For example, after learning a policy maximizing the rewards of one task, the agent starts exploring guided by the same policy as a second task is given. Transfer is task-centric rather than task-agnostic and environment-centric. Consequently, if tasks are too dissimilar information cannot be reused, even if the environments are similar. By contrast, in this work we propose learning task-agnostic exploration from one or many environments and show transfer to unseen environments. We should note that while Pathak et al. [120] did demonstrate fine-tuning on different maze maps, their focus and large-scale evaluations remain on tabula-rasa exploration.

3.3 Learning to Explore

Our goal is to decouple the environment-centric nature of exploration from its agent-centric component. Contrary to prior work, we propose to first learn an environment-centric exploration policy and then to transfer it to unseen environments. The policy is driven by the inherent interestingness of states and is learned over time via

interaction. First, during a pre-training phase, the agent interacts with many environments without any tasks and learns an exploration policy. Then, when new environments and tasks are presented, the agent uses the previously learned policy to explore more efficiently and learn task-specific policies. C-BET's key components are (1) a novel intrinsic reward and the learning of a policy to disentangle exploration from exploitation, and (2) the use of such policy to help exploration for new tasks. Figures 3.2 and 3.3 summarize our framework.

We should note that Rajendran et al. [128] also proposed a transfer framework based on intrinsic rewards. In their work, the agent switches between practice episodes –where the agent receives only intrinsic rewards– and match episodes –giving only extrinsic rewards. However, practice episodes are simpler variations of match episodes (e.g., in Atari Pong the agent practices against itself) rather than different tasks as in C-BET. Furthermore, the intrinsic reward used in practice episodes is given by a function trained with meta-gradients to improve the extrinsic-reward return. That is, exploration is not task-agnostic as in C-BET, and extrinsic rewards are the main drive of the agent.

3.3.1 Interestingness of State-Action Pairs

The natural world is filled with states or scenarios that are inherently interesting and our goal is to capture this inherent interestingness via intrinsic rewards. We propose adding an environment-centric component of interestingness to the existing agent-centric component of surprise. Specifically, we hypothesize that the environment can *change* on interaction, and the changes that are *rare* are inherently interesting. That is, we penalize actions not affecting the environment, and favor actions producing rare changes. For instance, moving around, bumping into walls, or trying to open locked doors without keys all result in no change and thus will be of low interest.

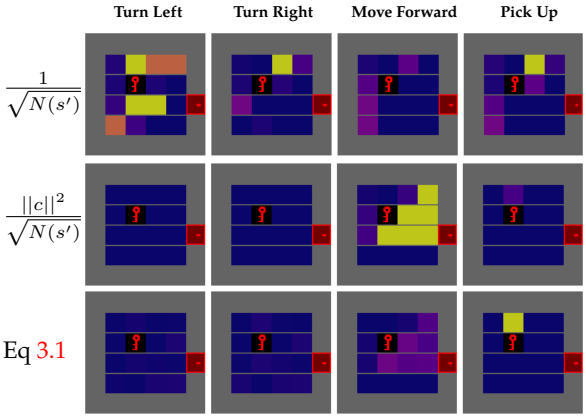
We also want to keep the agent-centric component in exploration –that is, the exploration policy should look for surprises or unseen states. Thus, we further reward actions leading to less-visited states. By combining these two components, the resulting C-BET interest-based reward is

$$r_i(s, a, s') = 1/(N(s') + N(c)), \tag{3.1}$$

where $c(s, s')$ defines the *environment change* of a transition (s, a, s'), and N denotes (pseudo)counts of changes and states. Figure 3.4 empirically shows its effectiveness. In Section 3.4 we discuss change encodings used in our experiments.

3.3.2 Exploration Learning

In this phase, we want to learn task-agnostic exploration policies from interaction with many environments. The agent has no goal, but states where it can 'die' are still



Gridworld with a key and a door. Observations encode cells depending on their content (e.g., 5 for the key, 10 for the agent). In each cell, the agent is facing downward and can pick up the key only from the cell above it. Samples have been collected randomly.

Figure 3.4: **Visualization of intrinsic rewards (row) for agent actions (column).** Brighter color denotes higher reward. Rewarding only state counts (top) does not provide useful feedback, and going to the corners is valued more than picking up the key. With the L2 norm of state changes (middle), the agent is biased in favor of moving, because its position is encoded with the highest value in the observation space. The resulting policy will prefer to navigate without picking up the key. In contrast, C-BET (bottom) gives picking up the key the highest reward.

terminal. In this setting, we would like to treat the problem of learning exploration as an MDP with intrinsic-rewards only, and train the agent to maximize discounted intrinsic-returns averaged over episodes.

Formally, the agent explores many environments $\mathcal{E}_{\text{EXP}} = \{E_1, E_2, \dots, E_N\}$, each governed by MDP $\langle S_n, A, P, r_i, \gamma_i \rangle$. That is, each environment has its own states but all environments have the same action space A, dynamics P, and intrinsic reward function r_i. The agent's goal is to learn an exploration policy maximizing the sum of discounted intrinsic rewards, i.e., $\pi_{\text{EXP}}(s, a) = \arg\max_\pi \mathbb{E}_{\mathcal{E},\pi}[\sum_t \gamma_i^t r_i(s_t, a_t)]$. To approximate the average, after a maximum number of steps the environment is reset and a new episode starts, as typically done in RL.

However, both common [21, 120, 127] and Eq. (3.1) intrinsic rewards decrease over time as the agent explores, to the point that they vanish to zero given enough samples. For instance, counts will grow to infinity, or prediction models error will go to zero. While this is not an issue in the tabula-rasa setup where the agent also gets extrinsic rewards, it can be problematic in the proposed task-agnostic exploration framework. Any policy, indeed, would be optimal if all rewards are zero.

To prevent Eq. (3.1) from vanishing, we randomly reset counts any given time step. To explain why resets need to be random, we start by considering 'episodic counts' proposed by Raileanu and Rocktäschel [127]. These counts are reset at the beginning of every episode to ensure that the agent does not go back and forth between a sequence of states with high rewards. While this work fine when extrinsic rewards are also given, it can be a problem if we learn only on intrinsic rewards. When counts are reset, the agent 'forgets' past trajectories and thinks that every state and change is new. If resets always happen at the end of an episode, then initial states will always get higher reward. Moreover, starting always with zero-counts may favor some trajectories and penalize others.

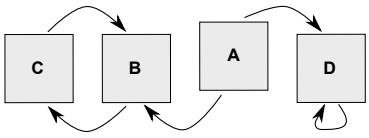


Figure 3.5: This chainworld illustrates that if counts are resets at the beginning of every episode, the learned policy will never visit D.

Consider the chainworld example in Figure 3.5. The agent always starts in A, from where it can go to B or D. From B, it loops between B and C. From D, it cannot go anywhere else. The optimal exploration policy should visit all states uniformly, by randomly going to B and D. However, if we reset counts at every episode, the agent forgets that it has already visited B and C. Thus the intrinsic rewards for B and C are high again, and trajectory ABCBCBC... gives higher intrinsic return than ADDD... Consequently, the optimal policy with respect to episodic counts will always prefer to visit B rather than D.

The optimal exploration policy, instead, should have some randomness to visit the environment uniformly, while prioritizing interesting states. For this reason, we propose to reset counts at any given step with probability p. When a new episode starts, counts may not be reset yet so the agent remembers what it has visited before. As the agent explores, on average common states and changes will have higher count more often, and the agent will correctly prefer rarer ones. In this work, we propose $p \leq 1 - \gamma_i$ where γ_i is the intrinsic reward discount factor. This is a fitting choice because in an MDP the sum of discounted rewards can be interpreted as the expected sum of undiscounted rewards if every time step had a $1-\gamma_i$ probability of ending. Intuitively, this means that γ_i implies a 'life expectancy' of $1/(1-\gamma_i)$ steps, and thus resets should not happen more frequently than that.

The resulting MDP with Eq. (3.1) rewards and random count resets can be solved by any RL algorithm. However, we note that this MDP is non-stationary, because the agent may receive different rewards for the same state, depending on how many times the state has been visited in the past. Nonetheless, classic intrinsic rewards –even in tabula-rasa exploration– either based on prediction errors [120, 127] or counts [12] also introduce non-stationarity because they change over time as well. In practice, this non-stationarity is not an issue because intrinsic rewards change slowly over time.

3.3.3 Exploration Transfer

Now, the agent is presented with new environments and asked to solve tasks. Formally, each environment is governed by the standard MDP $\langle S, A, P, r, \gamma \rangle$ and the agent's goal is to learn a policy that maximizes the sum of extrinsic rewards, i.e., $\pi_{\text{TASK}}(s, a) = \arg\max_\pi \mathbb{E}_\pi[\sum_t \gamma^t r(s_t, a_t)]$. Note that while during pre-training the policy was learned across all environments (one exploration policy for all environments), at transfer we learn one task-specific policy for each environment.

In this phase, the interest-policy learned earlier drives exploration as tasks and environments are presented to the agent. In order not to forget interestingness over time, the exploration policy is added as a fixed bias to the task-specific policy, sim-

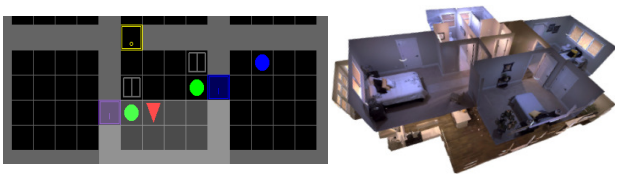


Figure 3.6: **Examples of the environments used in our experiments.** In MiniGrid (left), the agent navigates through a grid and interacts with objects (keys, doors, boxes, and balls) to fulfill a task. In Habitat (right), the agent navigates through visually realistic rooms.

ilarly to what Hailu and Sommer [70] proposed. Thanks to the decoupling of the interest-policy (based on the intrinsic reward) from the task-policy (based on the extrinsic reward), the latter can be also learned independently via any RL algorithm.

In experiments, we use IMPALA [54] to learn both π_{EXP} and π_{TASK}. IMPALA learns policies of the form $\pi(s, a) = \sigma(f(s, a))$, where σ is the softmax function. The policy is trained to maximize a function representing the value of states $V(s)$, trained on given rewards. In our framework, we combine the two policies as follows.

- During pre-training, by using intrinsic rewards we learn $V_i(s)$ and $\pi_{\text{EXP}}(s, a) = \sigma(f_i(s, a))$.
- At transfer time, we learn $V_e(s)$ on extrinsic rewards. The policy is $\pi_{\text{TASK}}(s, a) = \sigma(f_e(s, a) + f_i(s, a))$. The interestingness f_i is transferred but not trained, i.e., it acts as fixed bias to encourage interaction. Initially the policy follows f_i since f_e is initialized randomly. As it finds extrinsic rewards, the sum $f_e + f_i$ becomes greedier with respect to extrinsic rewards, and f_e slowly overtakes f_i[2].

Note that we transfer only f_i (the policy) and not V_i (the state value). We could think of transferring V_i as fixed bias as well, i.e., by having $V_e(s) = V(s) + V_i(s)$. The policy would be trained on V_e –the states value with respect to the given task– where V_i is fixed and only V is updated. However, we believe it is more beneficial to isolate the exploratory component within the policy, in order to keep the task-specific value function targeted on extrinsic rewards. By not transferring V_i, V_e can be accurately trained on extrinsic rewards –that the agent will see often thanks to f_i from the pre-trained policy. V_e, in turn, can make π_{TASK} greedy with respect to extrinsic rewards as V_e is learned.

3.4 Experiments

Our experiment design highlights the benefits of disentangling the environment-centric nature of exploration from agent-centric behavior by learning a separate exploration policy and then transferring it to new environments. We stress that for

[2]If exploration and the task goals are misaligned, we can decay exploration, e.g., $\pi_{\text{TASK}}(s, a) = \sigma(\alpha f_i(s, a) + f_e(s, a))$, where α decays over time, similarly to common ϵ-greedy policies.

learning task-agnostic exploration there are no standard benchmark environments, experimental setups, well-defined evaluation metrics, or even baselines to compare against. One of our contributions is to provide an exhaustive evaluation framework for the transfer exploration paradigm.

Environments. The experiments are divided into two main sections. The first is about MiniGrid [31] (Section 3.4.1), a set of procedurally-generated environments where the agent can interact with many objects. The second is about Habitat [138] (Section 3.4.2), a navigation simulator showcasing the generality of our MiniGrid experiments to a visually realistic domain.

Change encoding. In both MiniGrid and Habitat the agent partially observes the environment, since it cannot see through corners, closed door, or inside boxes, and has a limited field of view. Rather than egocentric views (i.e., what the agent sees in front of itself), we use 360° panoramic views to count environment changes, as this is a rotation-invariant representation of the observed state. Similar to Chaplot et al. [26], we concatenate four egocentric views taken from 0°, 90°, 180°, and 270° with respect to the North. Then, the change of a transition is the difference between panoramic views $\mathsf{pano}(s)$, i.e., $c(s, s') := \mathsf{pano}(s') - \mathsf{pano}(s)$.

Baselines. We evaluate against the following algorithms.

- *Count* [12]. The intrinsic reward is inversely proportional to the next state visitation count.
- *Random Network Distillation* (*RND*) [21]. The intrinsic reward is prediction error of states' random features between a trained network and a fixed one. This can be interpreted as similar to using state counts because the prediction improves states are seen more often.
- *Rewarding Impact-Driven Exploration* (*RIDE*) [127]. The intrinsic reward is prediction error between consecutive embedded states, normalized by episodic state counts.
- *Curiosity* [120]. The intrinsic reward is prediction error between consecutive states.

The source code is available at `https://github.com/sparisi/cbet/`.

3.4.1 MiniGrid Experiments

MiniGrid environments [31] are procedurally-generated gridworlds where an agent can interact with objects like keys, doors, and boxes (Figure 3.6). Exploration is challenging because rewards are sparse, observations are partial, and specific actions are needed to visit all states (e.g., pickup key to open door). With MiniGrid, we can generate several pairs of train and test environments that are related but still different in many ways. These pairs enable evaluation of both learning and transfer

abilities of an exploration method and its ability to deal with unseen components.

Implementation details. All environments give a $7\times7\times3$ partial observation encoding the content of the 7×7 tiles in front of the agent (including the agent's tile). The agent cannot see through walls, closed doors, or inside boxes. The action space is discrete: left, right, forward, pick up, drop, toggle, and done.

Setups. We present three setups, to study different exploration transfers against tabula-rasa.

- *MultiEnv* (*many-to-many transfer*). The agent loops over three environments one episode at a time and learns the exploration policy using intrinsic rewards only. There is one state count and one change count for all three environments rather than separate counts for each. The environments are: KeyCorridorS3R3, BlockedUnlockPickup, and MultiRoom-N4-S5, and have been chosen for size and interaction variety: the first has both a locked and an unlocked door, a key, and a ball; the second adds a box; the third has more rooms. Note that even if these environments have all object types, the agent cannot experience all kinds of interactions. For example, it will not know that keys can be hidden in boxes, as in the ObstructedMazes. The policy is then transferred to ten environments, seven of which are new. A good intrinsic reward should help learn better exploration faster from multiple environments, thanks to sharing experience from diverse interaction.

- *SingleEnv* (*one-to-many transfer*). The policy is pre-trained on a single environment. DoorKey and KeyCorridor are used for pre-training because they have some –but not all– objects.

- *Tabula-rasa* (*no pre-training / transfer*). A task-specific policy is learned as in classic intrinsic motivation by summing intrinsic and extrinsic rewards. While it is a non-realistic setup, it is the most common RL exploration approach, and thus serves as baseline against our transfer framework.

Evaluation metrics. Our goal is to learn exploration policies that encourage interaction with the environments and transfer well to new environments, i.e., that can further be trained to solve extrinsic tasks faster. Therefore, we evaluate policies according to the following criteria.

- Unique interactions across 100 episodes at transfer to new environments, after intrinsic-reward pre-training (no extrinsic-reward training yet). Unique interactions are picks/drops/toggles resulting in new environment changes. For instance, repeatedly picking and dropping the same key in the same cell results in only two interactions.

- Task success rate over 100 episodes at transfer to new environments, after intrinsic-reward pre-training (no extrinsic-reward training yet). The task success rate denotes in how many episodes the exploration policy visits goal states –thus, would have already solved the environment task.

- Extrinsic return during extrinsic-reward training, after intrinsic-reward training.

3.4.1.1 MiniGrid Pre-Training Results

Figure 3.7 shows results after pre-training in MultiEnv. C-BET policy both interacts with the environment and find goal states more often than all baselines. As we will see in the next section, this will result in faster extrinsic-reward learning.

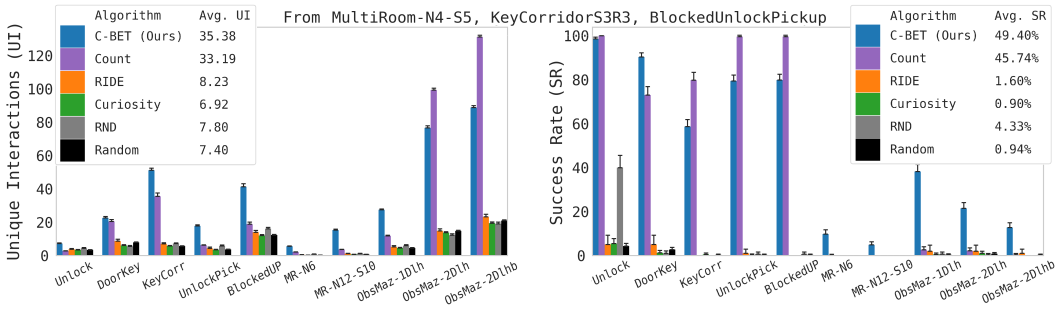


Figure 3.7: **Unique interactions and success rate** at the beginning of transfer of policies pre-trained in MultiEnv. Not only C-BET interacts the most and achieves the highest success rate, but also interacts and succeeds in **all** environments. Naturally, it interacts more in environment with many keys/balls/boxes to pick (KeyCorridor, BlockedUnblockPickup, ObstructedMazes), and less if there is nothing to pick (MultiRooms). On the contrary, Count overfits to the training environments and performs well only on the first five. Other baselines perform poorly, almost as a random policy.

Furthermore, C-BET's policy transfers well to all environments, even the ones with unknown dynamics (e.g., boxes in ObstructedMazes needs to be toggled to reveal keys). Of the baselines, only Count scores high average interactions and success rate, but it does not generalize as well as C-BET. Indeed, most of Count's success comes from environments visited at pre-training (the first five), but most of its interactions are in environments with unseen dynamics (ObstructedMazes). That is, Count's policy can explore familiar environments prioritizing state coverage (high success rate and few interactions), but not unfamiliar ones (low success rate yet high interactions).

Finally, RIDE, Curiosity, and RND baselines perform poorly. This is unsurprising if we consider that they rely on predictive models and that MiniGrid dynamics are deterministic and simple. Dynamics and embeddings models are learned quickly, without giving the policy time to explore.

3.4.1.2 MiniGrid Transfer Results

We transfer the exploration policies learned in Figure 3.7 as discussed in Section 3.3.3. Figure 3.8 shows how transfer setups (many-to-many and one-to-many) perform against tabula-rasa exploration.

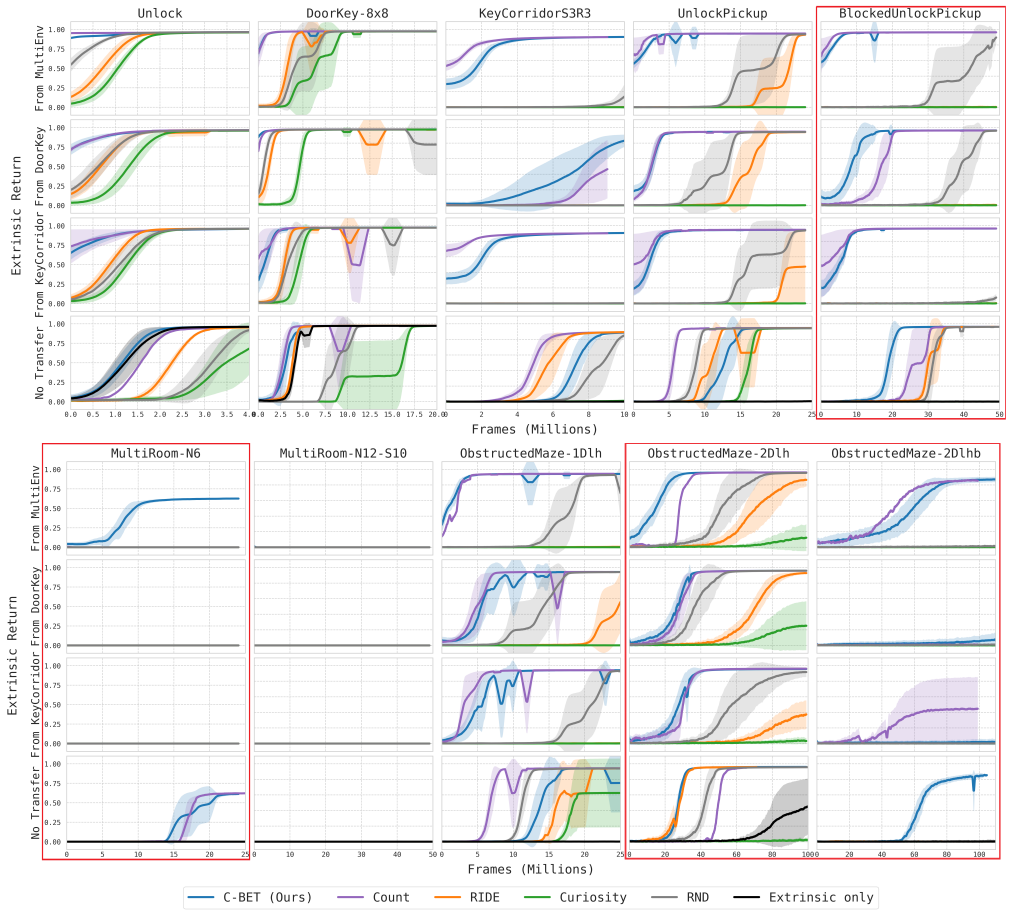


Figure 3.8: **MiniGrid task learning, for both transfer and tabula-rasa exploration.** The hardest tasks are outlined in red. C-BET (blue) from MultiEnv (top row under each environment) performs the best, starting with nearly optimal policies in most environments. This demonstrates the effectiveness of pre-training on multiple environments using the C-BET intrinsic reward.

The first takeaway is that policies pre-trained with the C-BET intrinsic reward outperform baselines in both transfer and tabula-rasa. In MultiEnv transfer, C-BET performs the best, especially on the hardest environments (outlined in red). In particular, only C-BET can transfer to MultiRoom-N6. On the contrary, Count –that can solve it in tabula-rasa– fails at transfer. C-BET is also the only solving ObstructedMaze-2Dlhb –the hardest environment of the ten– even in tabula-rasa.

The second takeaway is that baselines relying on models are not suited to the transfer framework. RIDE, Curiosity and RND perform better in the tabula-rasa setup (last row), except for the easiest environments (Unlock and DoorKey), meaning that transfer is actually harmful. These results are in line with Figure 3.7, where only C-BET and Count show success at offline transfer. Furthermore, RIDE, Curiosity and RND perform worst when transfer is from MultiEnv, highlighting that their intrinsic rewards are not suited for a multi-environment setup.

Finally, no algorithm learns MultiRoom-N12-10, not even C-BET despite showing some success in Figure 3.7. This is due to the randomly-initialized f_e of the task-specific policy, hindering the pre-trained exploration policy success.

3.4.2 Habitat Experiments

To demonstrate that C-BET's efficacy extends to realistic settings with visual inputs, we perform experiments on Habitat [138] with Replica scenes [153].

Implementation details. Egocentric views have resolution $64\times64\times3$. The action space is discrete: forward 0.25 meter, turn 10° left, and turn 10° right. To ease computational demands, we use #Exploration [158] with static hashing to map egocentric and panoramic views to hash codes and count occurrences with a hash table.

Setups. We evaluate Habitat on the *one-to-many transfer*. First, we pre-train exploration policies with only intrinsic rewards in one scene. Then, we evaluate them on new scenes without further learning. Given a fixed amount of steps, better policies will visit more of the new scenes.

Evaluation metrics. Unlike MiniGrid, we use no extrinsic rewards in Habitat. Since the agent has to navigate through rooms and spaces, we evaluate exploration policies using scene coverage measured by the agent's true state in Cartesian coordinates (not accessible by the agent)Faster, larger and more uniform coverage corresponds to better exploration. Plots show mean and confidence interval over seven random seeds per method with no smoothing.

3.4.2.1 Habitat Pre-Training Results

We pre-train exploration policies on Apartment 0 (Figure 3.6), one of the largest Replica scene in the dataset. Figures 3.9 and 3.11 show state coverage throughout and at the end of pre-training, respectively. C-BET explores more efficiently, covering twice as much of the scene than all baselines. In particular, at the end of pre-training it has explored almost all Apartment 0 uniformly.

3.4.2.2 Habitat Transfer Results

Here, we evaluate scene coverage of pre-trained policies in seven unseen scenes for episodes of fixed steps. A better exploration policy will exhibit generalization

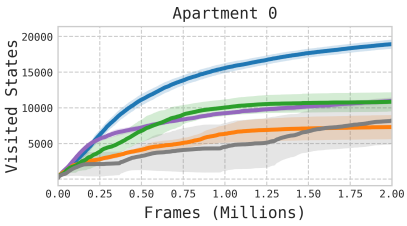


Figure 3.9: **Habitat pre-training.** C-BET explores the scene faster and scores the highest unique state count.

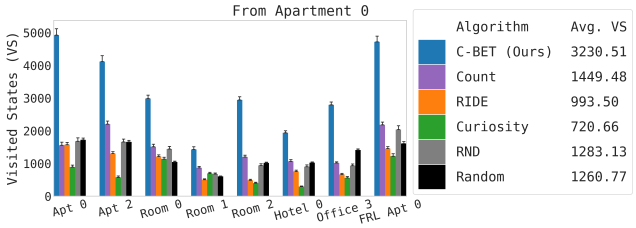


Figure 3.10: **Habitat offline transfer.** Bars denote the unique state count in an new scene during one episode. C-BET visits more than twice as many states than all baselines.

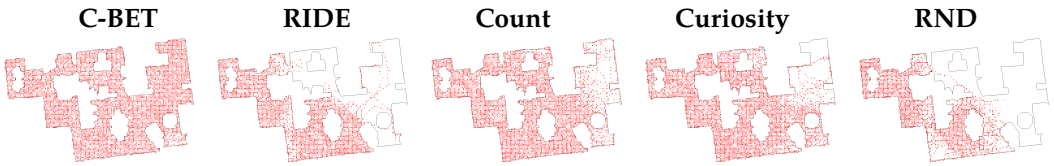


Figure 3.11: **Scene coverage** of exploration policies during pre-training (2M steps) in Apartment 0. Darker red cells denote higher visitation rates. Only C-BET visits all of the scene uniformly.

by covering a larger portion of all scenes as evenly as possible, an impressive feat given the visual complexity of the observations. Indeed, generalization is harder than MiniGrid because the lighting, colors, objects, and layout can be very different between scenes.Figures 3.10 and 3.12 show that, once again, C-BET clearly outperforms all baselines. Its exploration policy transfer well to all scenes, as it uniformly discovers more states. No baseline comes closer to its results. Actually, in many scenes baselines perform worse than a random policy.

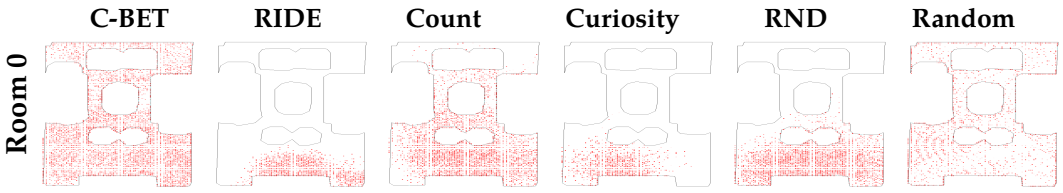


Figure 3.12: **Scene coverage** of exploration policies after 100 episodes (50,000 total steps) at offline transfer to Room 0. C-BET outperforms baselines and exhibits great transfer by visiting all of the scene uniformly.

3.5 Discussion

In this work, we proposed a paradigm change in task-agnostic exploration. Instead of studying task-agnostic exploration in isolated environments, we proposed to (1) learn task-agnostic exploration policies from one or multiple environments, and

(2) transfer learned exploration policies to unseen environments at testing time. In our setup, the agent interacts with the environment without any extrinsic goal and learns to explore environments in a task-agnostic manner. To this end, we proposed a novel intrinsic reward to encourage interaction with the environment and the visitation of unseen states. Subsequently, our agent effectively transfers its exploration policy to unseen environments.

Advantages. The proposed two-phase framework achieves two important features, making it fundamentally different from prior work. First, we account for *environment interestingness* without relying on additional models. Instead, we use a data-driven approach, estimating the rarity of states and environment changes. Rare changes are considered more interesting, actions causing them receive higher intrinsic rewards, and the agent is encouraged to perform them again. For instance, when navigating through rooms, opening doors will be more interesting due to rarity: the agent must navigate to the corresponding key, collect it, navigate to the door, and finally open it. Thus opening a door is rarer than picking up a key, in turn rarer than simple navigation movements. Furthermore, relying on environment-centric intrinsic rewards rather than task-centric extrinsic rewards facilitates learning from multiple environments at the same time. Second, contrary to prior transfer and continual learning algorithms we transfer policies learned on *interestingness of the environment* rather than task-specific policies. In the interest-based pre-training phase, we learn through interaction with the environment in a task-agnostic fashion, i.e., the agent freely explores the environment without any extrinsic task.

Limitations. In this work, we assumed that interacting with the environment while looking for rare changes helps find better extrinsic rewards faster. However, exploration and the task goals may be misaligned, thus a highly exploratory policy may slow down the discovery of extrinsic rewards. For instance, there may be dangerous states or harmful objects that the agent should avoid, even though they would make it curious during pre-training. Furthermore, C-BET is currently tied to (pseudo)counts to compute the rarity of states and changes. While extensions to continuous spaces exist, count-based metrics are more suited for discrete spaces.

Impact. RL can positively impact real-world problems, e.g., healthcare [66], assistive robotics [53], and climate change [135]. Yet, RL may have negative impacts, e.g., in autonomous weapons or workforce displacement [18]. Our work focuses on exploration in RL. Better understanding of what is interesting to do or visit helps exploration in unseen environments, as the agent will not waste time with random actions. Similarly, transferring policies learned in a related setting –as we do– can help narrow the range of the agent's expected behavior. Conversely, in many real-world scenarios exploration by curiosity and interestingness is unacceptable. For instance, autonomous cars cannot run over pedestrians just for the sake of curiosity. At present, our work is far from these impacts, but we hope to direct research to focus more on learning from multiple environments and transferring experiences, while at the same time ensuring the safety and reliability of autonomous agents.

Chapter 4

Hearing Touch: Audio-Visual Pretraining for Contact-Rich Manipulation

4.1 Introduction

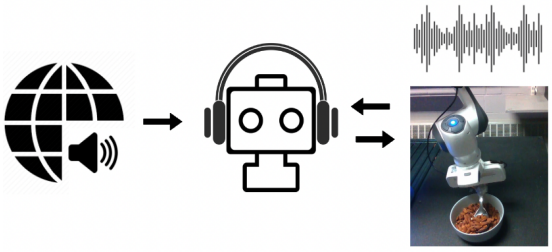


Figure 4.1: **Hearing touch**: We enable multisensory pretraining for manipulation by transferring audio-visual representations to robotic tasks using vision and contact audio.

Two key components consistently improve the performance of robotic manipulation: (1) pre-training on a large amount of data [51, 88, 99, 100, 108] and (2) using multisensory input, especially tactile sensing [22, 23, 93, 96, 106]. While recent work has leveraged pretraining on large-scale video datasets to create reusable *vision* representations for robot learning [99, 100, 108], there has been little focus on large-scale pretraining for other modalities such as tactile sensing. This gap arises due to the lack of relevant data at a comparable scale for tactile sensing. As a result, current approaches using non-visual sensory modalities are restricted to learning from a limited amount of task-specific data [93, 160]. How can we leverage internet data in pretraining tactile representations for manipulation?

Piezo contact microphones have emerged as a promising sensor in robotics due to their ability to capture high-frequency temporal information through structural vibrations captured as audio. Prior work has already demonstrated the ability to use contact audio for manipulation tasks [33, 96, 160]. In contrast to traditional tac-

tile sensors, the signal provided by contact microphones is inherently audio; hence recent work on learning audio-visual representations may apply to contact audio obtained from robot interactions.

We investigate how large-scale audio-visual training might be beneficial for learning contact audio representations for robotic manipulation. Our method makes use of Audio-Visual Instance Discrimination (AVID) [104], a self-supervised learning approach to learn audio-visual representations, pre-trained on Audioset [64], a dataset containing over 2 million human-labeled 10-second video clips of human and animal sounds, music, and environmental sounds drawn from the internet. Initializing our encoder with AVID weights, we train a policy with behavior cloning that fuses visual and audio inputs with self-attention in order to predict actions.

We validate our approach with experiments on three real-world manipulation tasks in the low-data regime, using at most 60 demonstrations per task. Surprisingly, despite the domain gap between the audio in Audioset and contact audio obtained through manipulation, we find that our approach improves performance over visual-only policies—especially in test settings where objects and locations differ significantly from the training data. Furthermore, our approach outperforms equivalent policies with audio encoders trained from scratch. Our experimental results reveal a promising avenue for multimodal pretraining across many robotic applications where neither vision alone nor training multisensory representations from scratch are sufficient.

4.2 Related Work

Audio in robotics Several works have shown the ability to reason over audio in robotics scenarios including object recognition [60], material classification, [35], estimating the volume and flow of granular material [33], exploration in RL [41], occluded manipulation [49], manipulation for sound replication [160], and waypoint setting in audio-visual navigation [28]. [96] introduce a mechanism for fusing input from a camera, a Gelsight sensor [171], and a contact microphone attached to the object of interest with self-attention for manipulation. Though our method also uses self-attention to fuse multisensory representations, we focus on leveraging large-scale audio pretraining, using visual input from a third-person camera and a contact microphone mounted directly on the robot. Our approach enables the robot to reason over vibrations caused by contact between tools and objects.

Tactile sensing for manipulation Several types of tactile sensors exist for application to robotic manipulation [91, 97, 14, 48, 156, 13]. We use contact microphones as an alternative tactile sensor, which are relatively inexpensive in comparison to common tactile sensors and can record vibrations with up to 1000 times higher frequency than other common tactile sensors (48000 Hz vs 30-400 Hz) [91, 97, 14]. Recent work has focused on applying traditional tactile sensors for learning to grasp

objects without visual observations [106] and in combination with visual observations for learning to improve the grasp of an object [23]. Our method using contact audio allows the sensor to measure vibrations directly via the sensor mounted on the gripper as well as indirectly through vibrations traveling along tools grasped by the gripper.

Audio-visual representation learning Self-supervised representation learning has been applied to the audio-visual domain, using audio-visual correspondence (AVC) as a form of cross-modal self-supervision from video [2, 3]. Other approaches have made use of the synchronization between vision and sound for sound representations [7], audio-visual sound separation [172], and sound localization [29]. More recent work has explored contrastive learning methods to discriminate between training instances using cross-modal and within-modal targets [104, 103, 122]. In our work, we use a pre-trained implementation of AVID [104] for obtaining audio-visual representations.

Representation learning for robotic manipulation

Several recent works have shown the benefit of using self-supervision to decouple representation learning of sensory inputs from behavior learning for robotic manipulation tasks [126, 160, 116, 93, 4]. A recent trend aims to obtain a universal visual representation—a single perception module pre-trained on large amounts of video data that can be frozen and used for downstream policy learning [99, 100, 108], however, there has been little focus on large scale pre-training for representation learning beyond vision in the context of robot manipulation. [160] also explores contact audio pre-training for behavior learning, however, their approach utilizes self-supervised learning using only task-specific data, whereas our method leverages the richness and diversity of large-scale audio-visual data for pre-training a contact audio representation. Further, we operate in the low-data regime with less than 100 demonstrations per task, whereas [160] collects 5,000 data points per task. We demonstrate the benefit of large-scale pre-training over SSL using only task-specific data in the low-data setting. To the best of our knowledge, our approach is the first to utilize large-scale multi-sensory representation learning for robotic manipulation.

4.3 Manipulation with Audio-Visual Pretraining

Given the difficulty and expense of collecting data in robotic settings, we turn toward leveraging more easily attainable large-scale sources of information such as internet data for learning manipulation policies. By utilizing contact microphones, we move beyond pre-training solely for visual input and obtain a means of pre-training a tactile sensor with large amounts of rich, audio-visual data. We outline

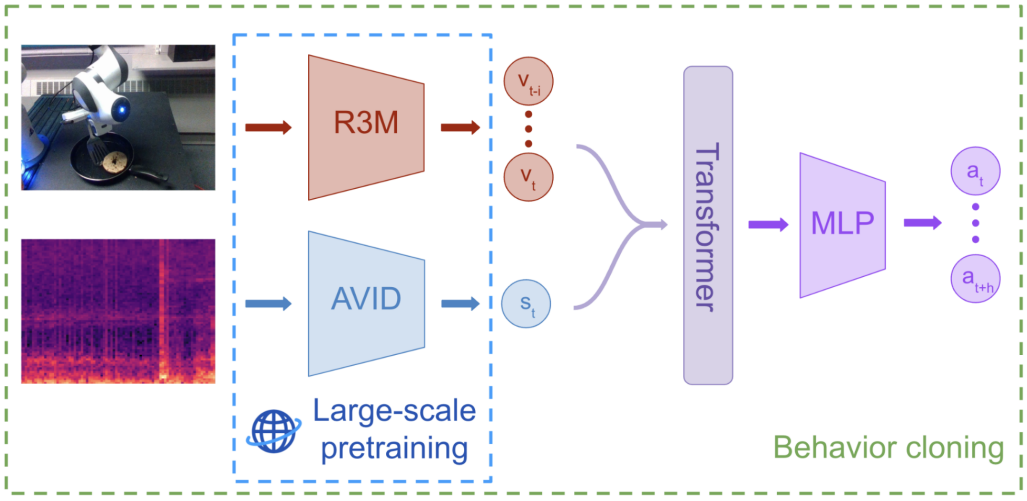


Figure 4.2: **Two-stage model training.** AVID and R3M pretraining leverages the large scale of internet video data (blue dashed box). We initialize the vision and audio encoders with the resulting pre-trained representations and then train the entire policy end-to-end with behavior cloning from a small number of in-domain demonstrations. The policy takes image and spectrogram inputs (left) and outputs a sequence of actions in delta end effector space (right).

further details of our approach in the following sections.

4.3.1 Sensors

At each timestep, we collect image observations o_t and two-second clips of contact audio a_t. The image observations are obtained from a third-person view camera and the audio is obtained by averaging the signal captured from four contact microphones mounted on the robot. Since contact microphones capture vibrations, they are sensitive to contact not only directly between objects and the sensors but also contact that results in vibrations traveling between objects. As a result, our setup allows the robot to sense subtle interactions between surfaces and tools that are grasped by the arm, as we show in the flipping task which requires the use of a spatula and the scooping task requiring the use of a spoon (Section 4.4.1).

4.3.2 Audio and Visual Representation Pretraining

Our method uses large-scale audio-visual pre-training to initialize our audio encoder and large-scale visual pre-training to initialize our visual encoder. The audio encoder is extracted from AVID [104] pre-trained on audio-visual pairs from Audioset [64] with cross-modal discrimination, encouraging the network to learn

video features that match the corresponding audio features and vice-versa. To isolate the effect of large-scale pre-training for our audio encoder, we use R3M [108], a proven method for pre-training visual features in robotic applications, R3M, with a ResNet18 [72] pre-trained on Ego4D human video dataset [68] with time contrastive learning and video-language alignment. Following [43], we keep both encoders unfrozen, continuing to update the weights during policy learning.

4.3.3 Audio-Visual Behavior Cloning

We train a policy with behavior cloning on a small number of in-domain demonstrations (described in Section 4.4.1). The model architecture is visualized in Figure 4.2. At each timestep, the policy takes in a two-second sound audio clip s_t and a sequence of i images $v_{t-i}, \ldots, v_t$ spanning the same two-second window, which are fed through the audio and image encoders, respectively. We apply learned positional embeddings to each of the five encoded representations and pass the result as input to a transformer decoder network similar to [96]. Similar to [32, 173] our method is quasi open-loop—the final component of our network is a multilayer perceptron that outputs the predicted delta actions $a_t, \ldots, a_{t+h}$ over a horizon of h timesteps. We optimize the network to minimize the standard MSE loss $\ell = \frac{1}{h}\sum_{j=0}^{h}(\Delta_{t+j} - \pi(v_{t-i}, \ldots, v_t, s_t)_j)^2$ averaged across all samples.

4.4 Experiments

In our experiments, we aim to answer two key questions: (1) Do contact microphones mounted on a robot arm capture interactions difficult to perceive with vision alone? (2) Does large-scale pre-training for audio-based tactile sensors yield representations that are useful for robot manipulation?

We answer these questions through real-robot experiments in the low-data setting. These experiments span three tasks (Section 4.4.2) and four methods (Section 4.4.3). We first describe our hardware setup (Section 4.4.1) before sharing implementation details. Our setup includes evaluation conditions requiring significant generalization beyond the training data.

4.4.1 Setup

Hardware We control a Franka Emika Panda Arm using an inverse kinematics solver to convert 6-DoF delta end effector Cartesian position and Euler rotation input to 7-DoF joint action. The end effector actions are commanded at 30 Hz. On the Franka gripper, we mount four Piezo contact microphones, each of which records audio at 32 kHz. We use an Intel D435 RealSense camera with a fixed third-person view to collect image observations at 30 Hz.

Figure 4.3: **Tasks and hardware setup.** We attach the Piezo contact microphones to our gripper (left), which record vibrations in the form of audio. We run experiments on three real-world tasks: scooping, spatula flipping, and zipping (right).

Data Collection Demonstrations are collected via teleoperation using an Oculus Quest headset. The visual data collected by the Intel D435 RealSense camera collects images with a resolution of 480×640. The audio waveforms from each of the four contact microphones are averaged across each sensor and downsampled to 16 kHz. We normalize the audio waveforms and generate mel spectrograms of the 2s audio segment, following the preprocessing of audio in [104].

4.4.2 Tasks

We present experiments on three real-world manipulation tasks, each described below. The scooping and zipping tasks are inspired by those in recent manipulation benchmarking efforts [40, 173]. The zipping task demonstrates the contact microphone's abilities to directly record vibrations touching the gripper, while the flipping and scooping tasks show their ability to record indirect contacts through vibrations traveling along tools (the spoon and spatula). Images of the three tasks are shown in Figure 4.3.

Flipping This task requires the robot to slide a spatula underneath an object, push the object to be "anchored" on the edge of the pan, and perform an upward motion that lifts one end of the object up and over onto its other side without pushing the object out of the pan. We collect 40 total demonstrations of flipping half of an upright bagel in a regular-sized pan (Figure 4.3). We then record the success rate of each model with the task of flipping four different objects, including three unseen during training, within a small pan, requiring the model to adjust to both the visual and physical differences due to the different objects and different pan (Figure 4.4a).

Scooping The robot is tasked with maximizing the weight of material scooped from a container using a spoon. The training and validation set consists of 60 demonstrations of scooping almonds out of a white bowl across three locations. We then

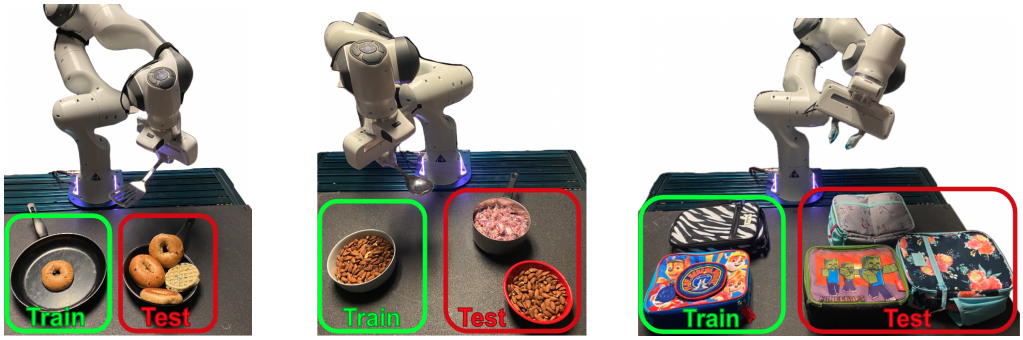
evaluate each model across two unseen locations, two types of scooping materials—including mints, unseen during training—and two different unseen containers (metal pot, red bowl) across three different joint configurations for each bowl and location for a total of 24 evaluations per model (Figure 4.4b). We record and report the average weight of the scooped material as well as the success rate (greater than 0 weight) across rollouts.

Zipping The robot begins the episode grasping the zipper of a lunchbox at the bottom left corner. The goal of the task is to entirely unzip the lunchbox, dragging the zipper along the edge of the lunchbox across three sides. In order to succeed the robot must pull the zipper in the right direction with the right amount of force and have the ability to turn corners which requires pulling the zipper in a rounding motion with a steady amount of force. We collect 50 total demonstrations split between two different lunchboxes and evaluate on three unseen lunchboxes (Figure 4.4c), recording the average distance zipped across all trials.

4.4.3 Baselines and Implementation Details

We conduct experiments with our method and three other baselines. We use different methods of pretraining in order to measure the effect of large-scale audio-visual pretraining on learning a useful contact audio representation for manipulation. All methods incorporating audio use the same audio encoder architecture, and all methods use R3M [108] pre-trained on Ego4d [68] with a ResNet18 [72] backbone as the initialization of the image encoder.

- **Vision-Only:** a baseline that shares the same architecture as our method, except that it only uses image frames as input. This baseline tests whether the signal from contact microphones is beneficial in our setup.
- **Scratch:** a baseline with randomly initialized weights for the audio encoder. This baseline tests how contact audio pretraining affects performance.
- **BYOL-A:** Bootstrap Your Own Latent for Audio (BYOL-A) [111], a self supervised approach to learning general-purpose audio representations that obtains multiple views of the same audio clip through augmentations and is trained to minimize the distance between these views in representation space. For each task, we train a model on the corresponding audio spectrograms for 100 epochs with a batch size of 1024, a learning rate of 0.0003, and the default settings for the network parameters and augmentations. We use the resulting network to initialize the audio encoder. This baseline compares the effect of large-scale audio-visual pre-training to task-specific audio pre-training, with an emphasis on the *amount* of pre-training data.



(a) Flipping task (b) Scooping task (c) Zipping task

Figure 4.4: **Train and Test Objects.** We perform experimental evaluations across a wide variety of objects with distinct visual differences in comparison with the objects used during data collection and training. The items on the left in each figure are the objects used for training and the objects on the right are used for evaluations.

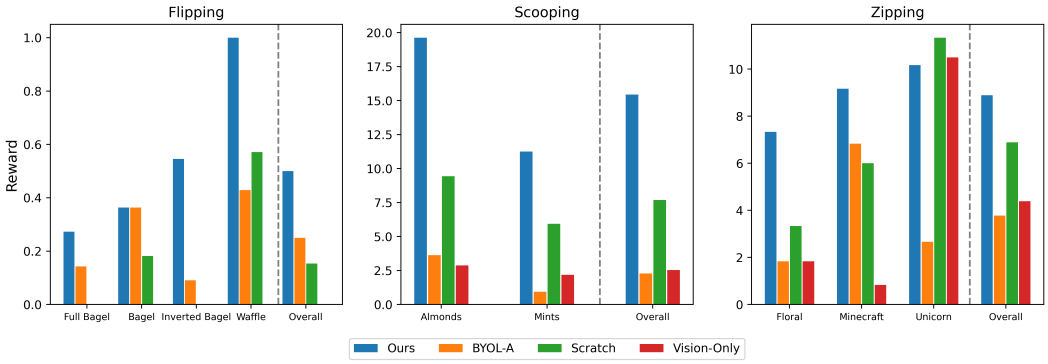


Figure 4.5: **Success rates across methods and tasks.** Our method, shown in blue, outperforms baselines in all but one setup of the zipping task. Furthermore, our method displays much less variation in performance between different configurations of each task, showcasing an increase in the ability to generalize to drastic visual differences as a result of learning useful audio representations.

We train all BC policies using a dropout probability of 0.5 and a batch size of 64 for a maximum of 100 epochs using early stopping with a patience of 15 epochs, choosing the model with the lowest validation loss. We use an Adam optimizer and a cosine annealing learning rate scheduler with a starting learning rate of 0.001. We train across three seeds and report the average results of each model across our three tasks. The different train and test objects are shown in Figure 4.4.

4.4.4 Results

The evaluation results across the different variations of each task are visualized in Figure 4.5 and summarized in Table 4.1. Our method using large-scale audio-visual pre-training outperforms all baselines across each of the three tasks with an average 23% higher 0-1 success rate and an average 76% increase in reward against the next best-performing baseline. Further, our method outperforms or matches the performance of all baselines in $8/9$ tasks, displaying a lower variation in performance between different configurations of each task, indicating greater robustness to visual features.

The Vision-Only baseline yields the worst performance across all tasks, providing evidence that contact audio improves the performance of manipulation policies over vision alone. Between BYOL-A and Scratch, the results are mixed—in the Flipping task BYOL-A outperforms Scratch and in Scooping and Zipping, Scratch performs better. Although BYOL-A includes an additional pre-training phase, the comparable performance with Scratch suggests that the augmentation techniques used by BYOL-A, while useful for learning audio representations for audio classification tasks when pre-trained on large audio datasets [111], are not effective when restricted to a small set of contact audio for learning manipulation policies. In contrast, our method utilizing AVID pre-training on Audioset greatly improves performance over Scratch and BYOL-A, demonstrating that the large-scale aspect of our method's audio-visual pre-training is the component most crucial to its success.

4.4.4.1 Qualitative Analysis

Many of the configurations of the task are difficult due to the noticeable visual differences between the train and test settings. As a result, the baselines suffer heavily from the domain shift and fail to generalize, often moving in jerk motions or away from the object of interest, even before coming into contact with objects. In contrast, our method appears to suffer less so from the significant visual differences,

Table 4.1: Average rewards and success rates across methods and tasks.

	Flipping	Scooping		Zipping	
	Success %	Reward	Success %	Reward	Success %
`Ours`	**50.0%**	**15.4**	**78.1%**	**8.9**	**88.9%**
`BYOL-A`	25.0%	2.3	25.0%	3.8	66.7%
`Scratch`	15.4%	7.7	50.0%	6.9	72.2%
`Vision-Only`	0.0%	2.5	28.1%	4.4	44.4%

suggesting that a good audio representation may prevent the model from overfitting to visual features during training.

The Vision-Only approach suffers most from the inability to perceive subtle interactions between surfaces, such as whether the spatula has successfully been slid under the bagel or the zipper is stuck on a corner. Despite having access to the same information as our method, the BYOL-A and Scratch baselines fail to reason effectively over the audio and utilize the additional information for taking actions.

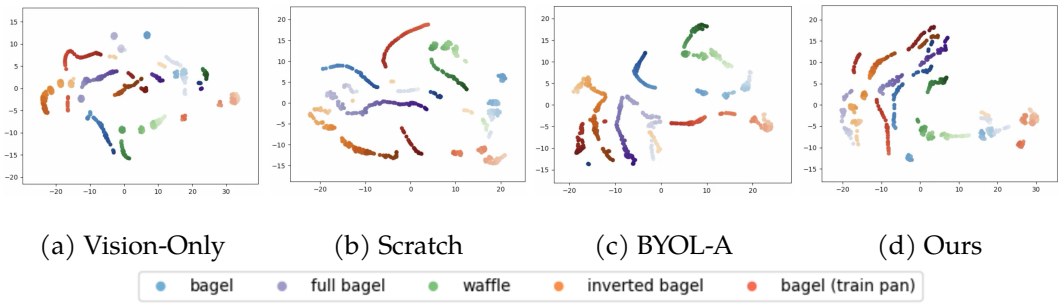
In the case of the scooping task, our method consistently learns to push the spoon deeper into the bowl until contact is made with the edge, and then tilt the spoon upward as the edge drags along the side of the bowl, increasing the amount of material scooped. This is more similar to the behavior of the training data in comparison with the baselines, which often fail to begin digging the spoon into the material as a result of misestimating the depth and relying on vision alone, or scoop too shallow.

4.4.4.2 t-SNE Visualizations

To better understand the learned representations of our method in comparison with the baselines, we visualize 2D projections of the transformer output embeddings using t-SNE initialized with PCA. For each method, we plot the projections of the embeddings from a sample trajectory over time for each variation of the flipping task, including both train and test settings. The visualizations are shown in Figure 4.6. For our method, although the representations are spaced apart at the beginning of the trajectories likely due to the visual differences across settings, the projections converge over the course of trajectories as the flipping motion is performed and completed. The visualization suggests the audio representations learned as a result of large-scale pre-training allow for the attention mechanism to better combine the audio-visual tokens, resulting in a more well-structured embedding space in comparison with the baselines.

4.5 Conclusion

In this chapter, we present a simple yet effective approach for improving manipulation performance by utilizing contact microphones as a tactile sensor. We argue that a primary strength of this sensor is that, in contrast to other sensors, it allows us to leverage large-scale internet data of the same modality and pretrain a representation that is useful for downstream robotic tasks. We show that the representations learned from large-scale audio-visual pretraining transfer well to such tasks despite the domain gap between contact audio in robotic manipulation and audio in internet videos. Future work could investigate which properties of pre-training datasets and objectives are most conducive to learning audio-visual representations for manipulation policies.



(a) Vision-Only (b) Scratch (c) BYOL-A (d) Ours

Figure 4.6: **t-SNE 2D projection.** For comparative analysis of the learned embedding spaces, we visualize projections of the learned representations from each method in each variation of the flipping task. Lighter hues indicate the starting points and darker hues indicate the end points of the trajectories. Please see the video on our website for a better visualization.

The lessons learned from our experiments echo those being shared across other machine learning subfields—more data is the driving factor in learning better models. Considering the safety issues, inefficiency, and resources required in collecting robotic data, it is unlikely that robotics will experience the scaling properties witnessed in data-rich domains such as natural language [17, 165]. Thus our goal is to widen the data scarcity bottleneck via methods that extract information from broader data sources that may be useful to an embodied agent.

4.6 Limitations

While contact microphones work well in our experiments, there are many tasks for which they may be less useful: less dynamic tasks such as pick and place, tasks with long periods of gripper movement in free space, situations where the robot itself generates vibrations that may affect the contact microphones or cases where the robot is working with deformable objects that do not emit perceptible vibrations upon contact.

Further, the gap between vision-only approaches in our experiments could be partially alleviated by equipping a robot with a wrist-based camera, although it may be impractical to modify the viewpoint of such a camera—especially in tasks involving tools. Successful robots of the future will likely be equipped with more than two types of sensors, and future work could develop policies that learn across several sensory modalities, performing active perception in order to increase the understanding of a situation or environment.

Chapter 5

Train Offline, Test Online: A Real Robot Learning Benchmark

5.1 Introduction

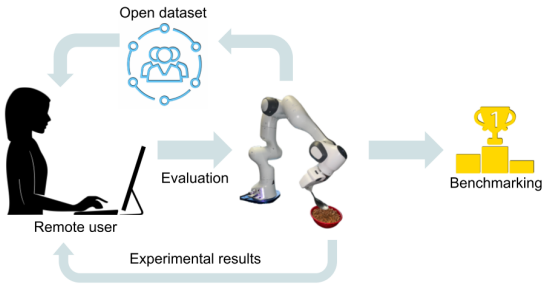


Figure 5.1: **Train Offline, Test Online**: Our benchmark lets remote users test offline learning methods on shared hardware.

One of the biggest drivers of success in machine learning research is arguably the availability of benchmarks. From GLUE [163] in natural language processing to ImageNet [45] in computer vision, benchmarks have helped identify fundamental advances in many areas. Meanwhile, robotics struggles to establish common benchmarks due to the physical nature of evaluation. The experimental conditions, objects of interest, and hardware vary across labs, often making methods sensitive to implementation details. Finally, the difficulties of purchasing, building, and installing infrastructure make it challenging for newcomers to contribute to the field.

For robotics research to advance, we clearly need a common way to evaluate and benchmark different algorithms. A good benchmark will not only be fair to all algorithms but also have a low participation barrier: setup to evaluation time should be as low as possible. Efforts like YCB [25] and the Ranking-Based Robotics Benchmark (RB2) [39] have aimed to standardize objects and tasks, but the onus of setting up infrastructure still lies with each lab. A simple way to overcome this is the use of a common physical evaluation site, as the Amazon Picking Challenge [37] and DARPA Robotics Challenges [19, 86, 145] have done. However, the barrier is

still high since participants must set up their own training infrastructure. Both of the above frameworks leave the method development phase unspecified and struggle to provide apples to apples comparisons.

Many robot learning algorithms do online training, where a policy is learned concurrently with data collection. One way to standardize online training is with simulation [16, 162, 170, 174]. While simulation mitigates issues with variation across labs, the findings from simulated benchmarks may not transfer to the real world. On the other hand, if we conduct online training in the real world, comparison across labs becomes difficult due to physical differences. In recent years, larger offline datasets have surfaced in robotics [36, 38, 101], and with them the rise of offline training algorithms. From imitation learning to offline reinforcement learning (RL), these algorithms can be trained using the same data and tested in a common physical setup.

Inspired by this observation, we propose a new robotics benchmark called **TOTO (Train Offline, Test Online)**. TOTO has two key components: (a) an offline manipulation dataset to train imitation learning and offline RL algorithms, and (b) a shared hardware setup where users can evaluate their methods now and going forward. Because all TOTO participants train using the same publicly-released dataset and evaluate on shared hardware, the benchmark provides a fair apples-apples comparison.

TOTO paves a path forward for robot learning by lowering the entry barrier: when designing a new method, a researcher can train their policy on our dataset, evaluate it on our hardware, and directly compare it to the existing baselines for our benchmark. TOTO means no more time devoted to setting up hardware, collecting data, or tuning baselines for one individual's environment. In this chapter, we lay out our benchmark design and present the initial methods contributed by benchmark beta testers across the country. These results show that our benchmark suite is challenging yet possible, providing room for growth as users iterate on TOTO.

5.2 Related Work

For a thorough description of work related to remote robotics benchmarking, we refer to the Robotics Cloud concept paper [42]. Here we describe related work specific to our instantiation of a robotics cloud (TOTO).

5.2.1 Shared Tasks and Environments

A necessary step in comparing method performance is evaluation on a common task. Common tasks might mean a standard object set such as YCB [25], which can be distributed to remote labs, allowing for shared metrics like grasp success on these objects. RB2 [39] provides four common manipulation tasks (similar to those we use, described in Section 5.3.2) as well as a framework for comparing and ranking

methods across results from multiple labs. Another route is sharing the environment itself, as the Amazon Picking Challenge [37] and DARPA Robotics Challenges [19, 86, 145] have done. Sharing tasks or environments gives metrics by which we can compare approaches. However, users must still develop the approach on their own hardware in their own lab, and recreating identical environment setups is quite challenging.

5.2.2 Shared, Remote Robots

Going one step further, remotely-accessible robots can be shared across the community, enabling method development and evaluation without users acquiring their own hardware. Georgia Tech's Robotarium [123] allows for remote experimentation of multi-agent methods on a physical robotic swarm, which has been extensively used not just in research but also in education. OffWorld Gym [87] provides remote access to navigation tasks using a mobile robot with closely mirrored simulated and physical instances of the same environment. A recent survey paper [155] provides an overview of robotic grasping and manipulation competitions, including some involving remotely-accessible, shared robots such as [98]. Finally, most closely related to our work, the Real Robot Challenge [58] runs a tri-finger manipulation competition on cube reorientation tasks. The success of the Real Robot Challenge inspires our work, which also allows for evaluation of manipulation tasks on shared robots. Our work, however, is designed to evaluate generalization in robot learning through challenging variations (lighting, unseen test objects, etc.) and an image-based dataset (as opposed to assuming ground-truth state access).

5.2.3 Open-Source Robotics Datasets

Collecting real-world robotics data is challenging and expensive due to physical constraints like environment resets and hardware failures. Thus open-source robotics datasets serve an important role in the field by enabling larger-scale offline robot learning. Some work has improved the way we collect robotics data, such as self-supervised grasping [124] and further parallelization of robots [95]. RoboTurk [101] provides a system for simple teleoperated data collection which can be executed remotely. Much work in robot learning has introduced datasets more generally, such as MIME [147] (8260 demonstrations over 20 tasks), RoboNet [38] (162,000 trajectories collected across 7 robots), and Bridge Data (7,200 demonstrations across 10 environments). However, it is hard to understand the value of these datasets without a common evaluation platform, something that [36] addresses by using simulation to replicate a real-world dataset. In contrast, we address this issue with real-world evaluation that matches the domain of the data collection. Our initial dataset is 2,898 trajectories, but this will grow over time as we add evaluation trajectories collected from users' policies.

5.2.4 Offline Robot Learning

TOTO focuses on offline robot learning, including imitation learning and offline RL. Our initial set of baselines is described and contextualized in Section 5.5.2.

5.3 The TOTO Benchmark

Our benchmark focuses on manipulation due to the lack of benchmarking in this area. Our hardware (Section 5.3.1) is set in environments that enable a set of benchmark manipulation tasks described in Section 5.3.2. We collect an initial dataset on these tasks, detailed in Section 5.3.3. Finally, in Section 5.3.4, we present the evaluation protocol for all policies contributed to our benchmark. For more information about our dataset and contributing to the benchmark, please see: https://toto-benchmark.org/.

5.3.1 Hardware

Our hardware includes a Franka Emika Panda robot arm and workstation for real-time inference. A simple joint position control stack runs at 30 Hz. The actions are joint targets, which are converted to motor control signals using a high-frequency PD controller. We also provide an end effector controller in which actions are specified via the position and orientation of the gripper. End effector control using X, Y, Z positions alone is not feasible to solve our tasks: for example, the orientation of the gripper must change as the robot pours. All the results presented in this chapter were attained using the joint position controller. We use an Intel D435 RealSense camera for recording RGB-D image observations.

We allow users to opt for a lower control frequency if desired. The training data can be subsampled by taking one of N frames since the actions are in absolute joint angles. We decrease the test time control frequency accordingly.

5.3.2 Tasks

The task suite consists of two manipulation tasks that humans encounter every day, similar to those introduced in prior work [10, 39]. The tasks are pouring and scooping, excluding the easiest and hardest RB2 tasks (zipping and insertion). Example image observations for these tasks are shown in Fig. 5.2. To see the original task designs, please refer to RB2: https://agi-labs.github.io/rb2/. Our tasks differ from those in RB2 in a few ways. We randomize the robot start state at the beginning of each episode. We apply a bit more noise to the target object locations. We use different combinations of objects based on availability. Lastly, we do not normalize the reward: the reward is the weight in grams of the material successfully scooped or poured. For detailed information on the task configurations, such as locations and objects, see our website.

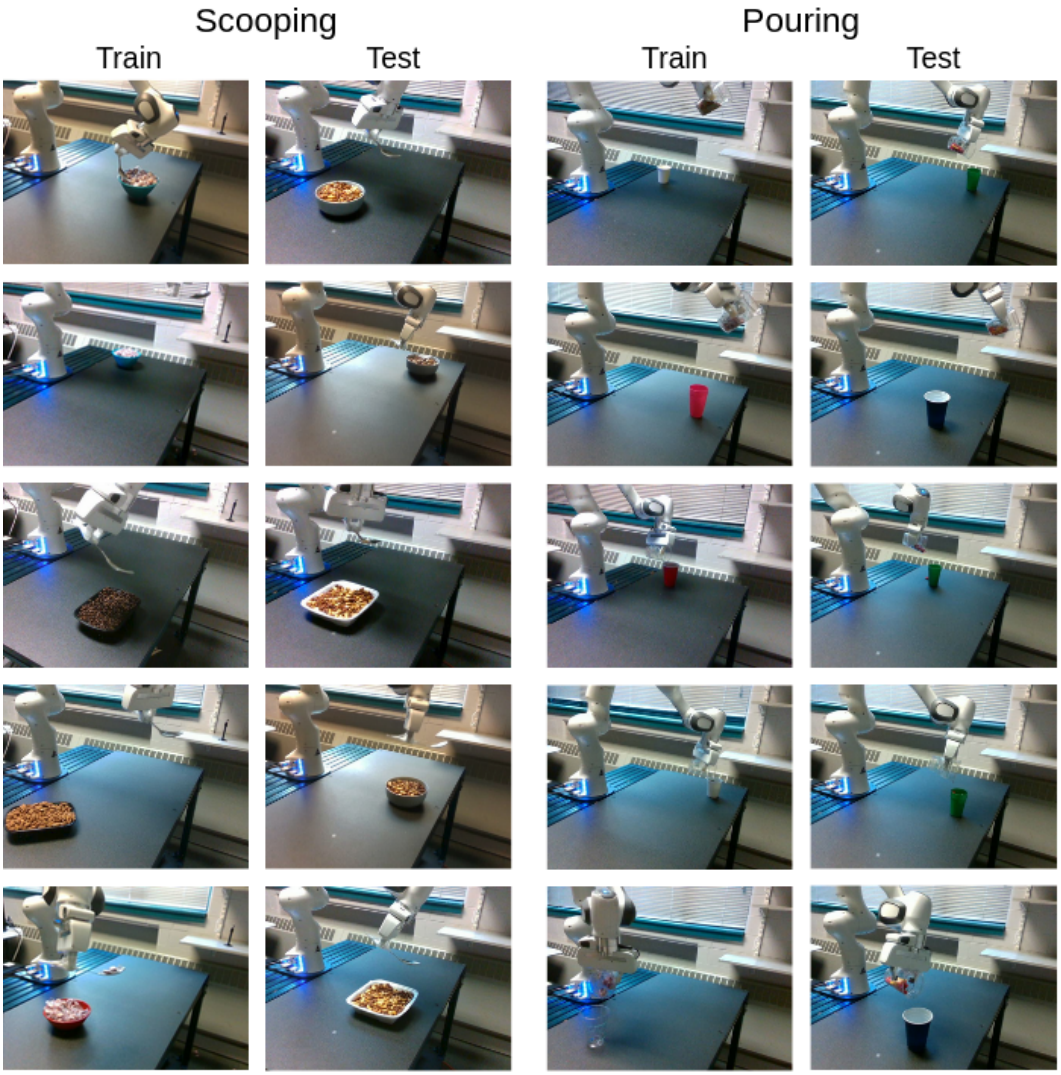


Figure 5.2: **TOTO Task Suite.** Our benchmark tasks are pouring and scooping, similar to those in RB2 [39]. Each involves challenging variations in objects, position, and more.

Scooping The robot starts with a spoon in its gripper and a bowl of material on the table. The objective is to scoop material from the bowl into the spoon. The training set includes all combinations of three target bowls, three materials, and six bowl locations (front left, front center, front right, back left, back center, and back right).

Pouring The robot starts with a cup containing granular material in its gripper. The goal is to pour the material into a target cup on the table. The training set includes all combinations of four target cups, two materials, and six target cup lo-

cations (same locations as scooping). The cup in the robot gripper is always clear plastic, enabling better perception of the material remaining in the cup.

5.3.3 Dataset

A key pillar of our benchmark is the release of a manipulation dataset. Dataset statistics (number of trials, average trajectory length, success rate, and data collection breakdown) are shown in Table 5.1. We consider a trajectory successful if it obtains a positive reward, and unsuccessful if the reward is zero. The initial release includes between 1000 and 2000 trajectories per task. Pouring data collection using replay and behavior cloning proved challenging to reset (unsuccessful trials require more cleanup), so it was nearly all collected with teloperation. Each trajectory includes images, robot actions (joint angle targets), joint states (joint angles), and rewards. To improve diversity, the data were collected with three techniques, each described below.

Table 5.1: Dataset overview.

	Task statistics			Collection technique		
	Trials	Length	Success	Teleop	BC	Replay
`Scooping`	1895	495	0.690	41%	33%	26%
`Pouring`	1003	324	0.977	99%	0%	1%

Teleoperation We collected the majority of trajectories with teleoperation using Puppet [89]. The human controls the robot in an intuitive end effector space using an HTC Vive virtual reality headset and controller. While this teleoperation is theoretically possible to use remotely, we collect the data with the human and robot in the same room, giving the human direct perception of the scene. Our multiple teleoperators have different dominant hands, leading to more diverse data. Most teleoperation trials are successful.

Behavior cloning rollouts After collecting teleoperation trajectories, we train simple, state-based behavior cloning (BC) policies on each target location, so no visual perception is required. We roll out BC trajectories with noise added to actions at each step. The amount of noise varies across trajectories for additional diversity.

Trajectory replay Finally, we also replay individual teleoperated trajectories with added noise. While these might seem overly similar to the original teleoperated trajectories, keep in mind that conditions like lighting also vary with time of day, so this replay still expands the dataset in other ways.

5.3.4 Evaluation Protocol

To evaluate each task, we use two unseen objects (bowls and cups) and one unseen material (mixed nuts for scooping and Starburst candies for pouring). We evaluate three object locations seen during training (front left, front center, front right) and three unseen locations. We evaluate three training seeds of each method. We initialize the robot with a randomly sampled pose at the beginning of each trajectory. However, the robot's initial poses are kept the same across seeds to ensure minimal variance. Combining 2 objects, 1 material, 3 locations, and 3 seeds means that each method is evaluated across 18 trials each for train and test locations. We report mean and variance of these trials.

5.4 Benchmark Use

Here we introduce the framework for our benchmark. TOTO is designed to make the user workflow (Section 5.4.1) easy for newcomers with well-documented software infrastructure (Section 5.4.2) including examples and tests.

5.4.1 User Workflow

We provide a real-world dataset (Section 5.3.3) collected using our hardware setup (Section 5.3.1). Participants optionally use our software starter kit (Section 5.4.2) and locally train policies of their choosing using this data. Users submit policies through Google Forms for evaluation on our real-world setup. They do not receive the low-level data from these evaluation trials; they simply receive a video showing the policy behavior as well as the reward and success rate.

An engineer supervises the real-world evaluations; thus evaluation turnaround time is currently around 12 hours (depending on time of day submitted). Our goal is to emphasize offline learning and prevent overfitting, removing the need for real-time results or large quantities of evaluation.

As new users evaluate methods after the TOTO release, we will post (anonymous) evaluation scores for each attempt on a website leaderboard. We will also periodically add data collected by the users' policies to the original dataset.

5.4.2 Software Infrastructure

Our software starter kit includes documented code and instructions for policy formatting and dataset usage. We have open-sourced baseline code, trajectory data, and pretrained models (see our website). These components ensure that TOTO is easily accessible to a broad portion of the robotics, ML, and even computer vision communities.

We adapt the agent format from [80], which requires a `predict` function taking in the observation and returning the action. We use a standard config format and require an `init_agent_from_config` function to create the agent.

We provide users with code for training an example image-based BC agent and a docker environment which wraps the minimum required dependencies to run this code. Users can optionally extend the docker containers with additional dependencies. We also provide a stub environment for users to locally verify whether their agent's predictions are compatible with our robot environment. This setup allows users to resolve all agent format and library dependency issues before submitting agents for evaluation.

5.5 Baselines

We highlight the importance of establishing a benchmark by running two sets of experiments: (a) what is a good visual representation for manipulation? and (b) what is a good offline algorithm for policy learning? To test the benchmark infrastructure, we have solicited baseline implementations for both experiments from several labs.

5.5.1 Visual Representation Baselines

A core unanswered question, due to the lack of benchmarking, is what is a good visual representation for manipulation? Is ResNet trained on ImageNet great or do self-supervised approaches outperform supervised models? We evaluate five visual representations provided by TOTO users from multiple labs. Two are trained on our data (in-domain) and three are generically pretrained.

`BYOL` (Bootstrap Your Own Latent) [69] is a self-supervised representation learning method trained on our dataset. The BYOL representation embedding size is 512.

`MoCo (Generic)` refers to the Momentum Contrast (MoCo) model trained on ImageNet [73], while `MoCo (In-Domain)` is trained on our data with crop-only augmentations [118].

`Resnet50` is trained with supervised learning on ImageNet [72].

`R3M` (Reusable Representations for Robot Manipulation) [108] is trained on Ego4D [68] with time-contrastive learning and video-language alignment. For `R3M`, `MoCo`, and `Resnet50`, we use the 2048-dimensional embedding vector following the fifth convolutional layer.

These representations performed the best among a larger set of vision models on which we ran an initial brief analysis (including offline visualizations and BC rollouts). Additional representations that performed less well (and therefore are not included as baselines) included CLIP [125] and a lower-level MoCo model (from the third layer instead of the fifth).

5.5.2 Policy Learning Baselines

Remote users have contributed the below policy learning baselines, which span the spectrum from nearest neighbor querying to BC to offline RL. They were selected according to each TOTO contributor's expertise with approach coverage in mind. All methods use RGB image observations, and some run these images through a frozen, pretrained vision model before passing the resulting embedding to a policy.

`BC` is trained on top of each vision representation baseline. Closed-loop BC predicts a new action every timestep, while open-loop BC predicts a sequence of actions to execute without re-planning. Our BC baseline is *quasi* open-loop: training trajectories are split into 50-step action sequences, and the policy is trained to predict such a sequence given the current observation. During evaluation, these 50 actions are executed between each prediction step. We find that this performs better than closed-loop or open-loop alone: closed-loop struggles without history, and open-loop is challenging with our variable-length tasks. We filter the training data to only include trajectories with nonzero reward [30].

`VINN` (Visual Imitation through Nearest Neighbors) [116] is a nearest neighbor policy using an image encoder trained with `BYOL` [69]. While using nearest neighbors as a policy has been previously explored [102], this approach alone does not scale well to high-dimensional observations like images. BYOL maps the high-dimensional observation space to a low dimension to obtain a robust policy. VINN was originally closed-loop (query and execute a new action at each timestep), but in this work we mirror the 50-step quasi open-loop approach used in the BC baseline (described above).

`IQL` (Implicit Q-learning) [85] uses the open-source implementation from the `d3rlpy` package [146]. We use `MoCo (In-Domain)` as a frozen visual representation since it performed the best in our comparison of representations with BC. We concatenate the frozen image embeddings with the robot's joint angles as the input state to the model.

`DT` (Decision Transformers) [30] recasts offline RL as a conditional sequence modeling task using transformers. Similar to BC, it is trained to predict the action in the dataset, but conditions on the trajectory history as well as target return (desired level of performance). We use the Hugging Face DT implementation. The model receives an RGB image and the robot's joint angles: the former is embedded using `MoCo (In-Domain)` and concatenated with the latter at each step. DT uses a sub-sampling period of 8 and a history window of 10 frames. For inference and evaluation, the target return prompt is approximately chosen as the mean return from the top 10% of trajectories in the dataset for each task.

5.6 Experimental Results

5.6.1 Visual Representation Comparison Using BC

Our first set of experiments compares BC agents using the vision representations detailed in Section 5.5.1 and evaluated with the protocol described in Section 5.3.4. The success rates across all representations and tasks are visualized in Fig. 5.3, and the numerical rewards are presented in Table 5.2.

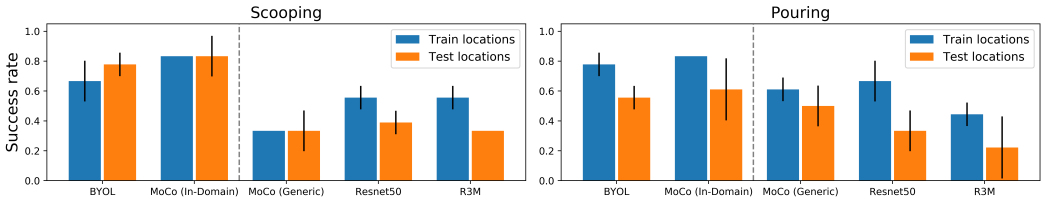


Figure 5.3: **Vision representation comparison with BC.** Models trained on our data (left of dashed line) perform better than generic ones (right of dashed line), and results tend to be better for training object locations than unseen test locations.

Table 5.2: Performance of vision representations with BC across all locations.

	Method	Scooping		Pouring	
		Reward	Success %	Reward	Success %
In Domain	`BYOL`	4.39	72.2%	20.22	66.6%
	`MoCo`	**7.42**	**83.3%**	**22.86**	**72.2%**
Out of Domain	`MoCo`	2.11	33.3%	14.89	55.5%
	`ResNet50`	2.83	47.2%	18.86	50.0%
	`R3M`	2.97	44.4%	6.94	33.3%

These results show that finetuning the MoCo model on our data outperforms the generic version, as expected. `MoCo (In-Domain)` achieves the highest success rate and average reward on both tasks, followed by `BYOL`, the other in-domain model. In general, the relative performance between models is mostly consistent across tasks. Resnet50 and `MoCo (Generic)` perform slightly better on pouring than on scooping.

Fig. 5.3 also visualizes performance differences due to object locations. Locations seen during training perform better, but performance does not degrade significantly, suggesting that the representations have a generalizable notion of where the target object is. Surprisingly, the two representations trained on our data (`MoCo`

(In-Domain) and BYOL) perform equally good or even slightly better on unseen locations for scooping.

5.6.2 Policy Learning Results

Table 5.3 shows the comparison of policy learning methods (described in 5.5.2) evaluated on TOTO. Due to compute constraints, we have 1 and 2 seeds for DT and IQL respectively. We compensate by duplicating the evaluation of these seeds to keep the number of trials consistent. The results are visualized in Fig. 5.4. We find that VINN performs the best in train locations. We also note that offline-RL approaches (especially IQL) achieve some success unlike in RB2[39]. This is likely due to a larger and more diverse dataset than RB2, which contributes to better offline RL performance.

Table 5.3: TOTO policy learning results across train and test locations.

Method	Scooping		Pouring	
	Reward	Success %	Reward	Success %
BC + MoCo	7.42	**83.3%**	**22.86**	**72.2%**
VINN	**7.89**	63.9%	21.75	47.2%
IQL	6.08	47.2%	9.86	38.9%
DT	2.83	27.8%	0.00	0.0%

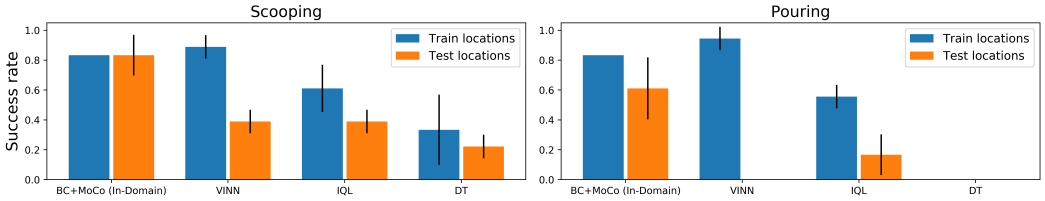


Figure 5.4: **Evaluating offline policy learning results.** VINN sees the best performance on train locations, but its performance degrades on unseen locations, as does the performance of other methods.

We found that scooping proves challenging due to a non-markovian aspect of the task: the spoon is above the bowl both before and after scooping. Thus we would expect open-loop methods (BC, VINN) and those with history (DT) to perform better than others in this setting. While BC and VINN achieve competitive performance on scooping, DT only achieves moderate success on scooping and does not see any positive rewards on pouring. Meanwhile, IQL provides decent performance without history on a non-markovian task.

Comparing the train and test location results for policy learning proves interesting. VINN performs the best on train locations, but it struggles on unseen locations, since it selects actions using the nearest neighbor trajectory from the training data. All other methods also experience some level of degradation when moving to unseen locations, leaving one clear direction for method improvement using TOTO.

5.6.3 Dataset Size Ablation

To understand the impact of dataset size on policy learning performance, we perform an ablation in which we train BC on the scooping task with varying amounts of data. We sort the scooping trajectories by reward and train policies with the top 5%, 25%, 50%, 100% of the data, as well as all successful trajectories with positive rewards (∼70%). This sorting by reward ensures that policies trained in the small-data regime are not overcome by unsuccessful trajectories. We present the dataset size ablation results in Table 5.4.

Table 5.4: Dataset size ablation with BC.

Dataset size	Reward	Success %
5%	2.89	38.9%
25%	5.94	72.2%
50%	6.22	77.8%
Successes (∼70%)	8.06	83.3%
100%	5.00	72.2%

The *all success* number uses the same policy as the BC policy in Table 5.3, but we evaluate it again with the ablations to ensure minimal variance in conditions. As expected, training on more data generally leads to a higher success rate. Training on all of the data (including unsuccessful trajectories) leads to a lower reward than training on only the successful trajectories, also unsurprising given the use of BC to learn the policies in this ablation (we might expect offline RL to improve with the inclusion of unsuccessful trials).

Overall, these ablation results suggest that the TOTO dataset size is the right order of magnitude in terms of policy learning. We have reached the point of diminishing returns: training on 50% versus 70% of the data does not substantially improve performance. However, additional data might still improve visual representation learning.

5.6.4 Metrics for Offline Policy Evaluation

A TOTO user might wish to sanity check their policy before submitting it for real-world evaluation or otherwise have performance metrics to guide offline tuning. Here we present simple example metrics for offline evaluation: action similarity to a validation set of expert demonstrations using both joint angle error and end effector pose error. From a chosen validation set of 100 trajectories, we estimate

the joint angle error and end effector error by computing the mean squared error between agent's predicted actions and actual actions for all samples.

Fig. 5.5 shows these validation metrics on BC checkpoints throughout training and the real-world reward evaluated on four representative checkpoints. The reward increases as the validation error metrics decrease, matching expectations. These metrics capture overfitting: the overtrained policy from 2,000 epochs shows a significant decrease in real-world reward and likewise has higher validation error. While offline metrics alone should not fully guide the development of an algorithm, it provides a signal as to whether the policy might achieve any success in the real world.

Figure 5.5: **Comparing offline evaluation to online performance.** While offline evaluation is imperfect, it provides a sanity check to the user, guiding development at a higher frequency than real-world evaluation.

5.7 Discussion

The main goal of this work is to introduce TOTO, our robotics benchmark. We presented a broad initial set of baselines containing both vision representations and policy learning approaches, which can be built off of by future TOTO users. Notably, these baselines were contributed in the same way that TOTO will be used in the future: by collaborators who locally train policies and submit them for remote evaluation on shared hardware. This shows the feasibility of our user workflow. The initial baseline results show the challenging nature of our tasks, especially with respect to generalization. By using TOTO as a community, we can more quickly iterate on ideas and make progress on the real-world bottlenecks to robot learning.

5.7.1 Limitations and Future Work

The evaluation protocol currently has manual steps: we measure the material transferred during pouring and scooping to compute rewards and reset by returning the material to the original object. We do see future potential to automate reward measurements and resets, such as by adding a scale beneath the target object and using an additional robot to reset the transferred materials. Spills of the transferred material, however, might still require manual intervention.

We plan to expand the evaluation setup to include additional robots. This would

help us meet the increasing demand in evaluations as more users adopt the benchmark. One challenge will be visual differences across robots, but we plan to collect additional demonstrations on new robots, and this would be an opportunity to expand the set of tasks as well (we could designate one robot per task).

As user demand further grows, we will implement an evaluation job queue which prioritizes evaluation requests from different users and schedules the jobs based on the number of robots currently available.

Chapter 6

Outlook

The work presented in this thesis improves robot learning for real-world practicality through two directions: using prior data and shared evaluation. First, I have improved training efficiency by leveraging multimodal richness, transferring knowledge from past environments, and using cheap passive data. These improvements together enable an agent to learn quickly in new environments, mitigating the amount of costly trial-and-error data required. Second, the TOTO benchmark for shared evaluation has the aim of simultaneously accelerating technical research and democratizing contributions to robotics.

I am excited to continue improving robot learning, especially by extending TOTO. The TOTO benchmark is a prototype of a larger vision for shared evaluation: the Robotics Cloud, a center filled with dozens (if not hundreds) of real, remotely operable robots on which any researcher can run experiments, collect data, and benchmark their algorithms. TOTO has shown that remote evaluation is possible for robotics researchers. Scaling TOTO to the full vision means enabling remote data collection, perhaps teleoperation, and scaling the hardware to accomodate a larger capacity.

More than solving a technical problem of benchmarking algorithms, the Robotics Cloud effort also approaches a larger problem in the robotics community: making the field accessible. The TOTO infrastructure we have developed makes getting started easy: in fact, three of five initial TOTO submissions had undergraduate contributors. This shows promise for making the Robotics Cloud accessible to not only those at elite research institutions but also those without the expertise or funding to work directly with robots (such as vision researchers or even high school students). I see the Robotics Cloud bringing another kind of richness to robotics: a diverse set of people and ideas.

Bibliography

[1] J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *CoRR*, abs/1703.01732, 2017. 5

[2] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017. 34

[3] R. Arandjelovic and A. Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018. 34

[4] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. *arXiv preprint arXiv:2203.13251*, 2022. 34

[5] P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 49–56, 2007. 19

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002. 19

[7] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29, 2016. 34

[8] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, 2016. 6

[9] Y. Aytar, T. Pfaff, D. Budden, T. Paine, Z. Wang, and N. de Freitas. Playing hard exploration games by watching youtube. In *Advances in Neural Information Processing Systems*, 2018. 6, 9, 13

[10] S. Bahl, A. Gupta, and D. Pathak. Hierarchical neural dynamic policies. *arXiv preprint arXiv:2107.05627*, 2021. 46

[11] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. Van Hasselt, and D. Silver. Successor features for transfer in reinforcement learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 20

[12] M. G. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 5, 7, 17, 19, 23, 25

[13] T. Bhattacharjee, A. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp. Tactile sensing over articulated joints with stretchable sensors. In *2013 World Haptics Conference (WHC)*, pages 103–108. IEEE, 2013. 33

[14] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021. 33

[15] R. I. Brafman and M. Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 3 (Oct):213–231, 2002. 19

[16] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. 2016. 44

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 42

[18] E. Brynjolfsson and T. Mitchell. What can machine learning do? workforce implications. *Science*, 358(6370), 2017. 16, 31

[19] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. Springer, 2009. 43, 45

[20] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2018. 4, 6, 7, 10, 11, 12

[21] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019. 7, 11, 19, 22, 25

[22] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017. 32

[23] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. 32, 34

[24] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4), 2018. 6

[25] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics*, pages 510–517. IEEE, 2015. 43, 44

[26] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12875–12884, 2020. 25

[27] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6, 11, 14

[28] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, 2020. 33

[29] H. Chen, W. Xie, T. Afouras, A. Nagrani, A. Vedaldi, and A. Zisserman. Localizing visual sounds the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16867–16876, 2021. 34

[30] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34:15084–15097, 2021. 51

[31] M. Chevalier-Boisvert, L. Willems, and S. Pal. Minimalistic Gridworld Environment for OpenAI Gym, 2018. URL https://github.com/maximecb/gym-minigrid. 25

[32] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 36

[33] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer. Learning audio feedback for estimating amount and flow of granular material. *Proceedings of Machine Learning Research*, 87, 2018. 32, 33

[34] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer. Learning audio feedback for estimating amount and flow of granular material. In *Conference on Robot Learning*, 2018. 6

[35] S. Clarke, N. Heravi, M. Rau, R. Gao, J. Wu, D. James, and J. Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *Conference on Robot Learning*, pages 662–673. PMLR, 2022. 33

[36] J. Collins, J. McVicar, D. Wedlock, R. Brown, D. Howard, and J. Leitner. Benchmarking simulated robotic manipulation through a real world dataset. *IEEE Robotics and Automation Letters*, 5(1):250–257, 2019. 44, 45

[37] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 15(1):172–188, 2016. 43, 45

[38] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019. 44, 45

[39] S. Dasari, J. Wang, J. Hong, S. Bahl, Y. Lin, A. S. Wang, A. Thankaraj, K. S. Chahal, B. Calli, S. Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 43, 44, 46, 47, 53

[40] S. Dasari, J. Wang, J. Hong, S. Bahl, Y. Lin, A. Wang, A. Thankaraj, K. Chahal, B. Calli, S. Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. *arXiv preprint arXiv:2203.08098*, 2022. 37

[41] V. Dean, S. Tulsiani, and A. Gupta. See, hear, explore: Curiosity via audio-visual association. *Advances in Neural Information Processing Systems*, 33:14961–14972, 2020. 2, 33

[42] V. Dean, Y. G. Shavit, and A. Gupta. Robots on demand: A democratized robotics research cloud. In *Conference on Robot Learning*, pages 1769–1775. PMLR, 2022. 2, 44

[43] V. Dean, D. K. Toyama, and D. Precup. Don't freeze your embedding: Lessons from policy finetuning in environment transfer. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022. URL https://openreview.net/forum?id=HBHMrQD-LZc. 36

[44] W. N. Dember and R. W. Earl. Analysis of exploratory, manipulatory, and curiosity behaviors. *Psychological Review*, 64(2), 1957. 5

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 43

[46] C. D'Eramo, A. Cini, and M. Restelli. Exploiting Action-Value uncertainty to drive exploration in reinforcement learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2019. 19

[47] K. Dong, Y. Wang, X. Chen, and L. Wang. Q-learning with UCB exploration is sample efficient for Infinite-Horizon MDP. In *International Conference on Learning Representation (ICLR)*, 2020. 19

[48] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018. 33

[49] M. Du, O. Y. Lee, S. Nair, and C. Finn. Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning. *arXiv preprint arXiv:2205.14850*, 2022. 33

[50] R. Dubey, P. Agrawal, D. Pathak, T. L. Griffiths, and A. A. Efros. Investigating human priors for playing video games. In *International Conference on Machine Learning (ICML)*, 2018. 17

[51] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 32

[52] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune. Go-Explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019. 13

[53] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp. Assistive gym: A physics simulation framework for assistive robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10169–10176. IEEE, 2020. 31

[54] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018. 24

[55] F. Fernández and M. Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2006. 20

[56] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017. 20

[57] R. Flom and L. E. Bahrick. The development of infant discrimination of affect in multimodal and unimodal stimulation: The role of intersensory redundancy. *Developmental psychology*, 43(1), 2007. 5

[58] N. Funk, C. Schaff, R. Madan, T. Yoneda, J. U. De Jesus, J. Watson, E. K. Gordon, F. Widmaier, S. Bauer, S. S. Srinivasa, et al. Benchmarking structured policies and policy optimization for real-world dexterous object manipulation. *arXiv preprint arXiv:2105.02087*, 2021. 45

[59] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020. 6

[60] D. Gandhi, A. Gupta, and L. Pinto. Swoosh! rattle! thump!–actions that sound. *arXiv preprint arXiv:2007.01851*, 2020. 33

[61] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6

[62] R. Gao, C. Chen, Z. Al-Halab, C. Schissler, and K. Grauman. Visualechoes: Spatial image representation learning through echolocation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6

[63] J. Garcıa and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1), 2015. 16

[64] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 33, 35

[65] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 8

[66] O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18, 2019. 31

[67] J. Gottlieb, P. Oudeyer, M. Lopes, and A. Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in Cognitive Sciences*, 17(11):585–593, 2013. 19

[68] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Conference on Computer Vision and Pattern Recognition*, pages 18995–19012. IEEE, 2022. 36, 38, 50

[69] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 50, 51

[70] G. Hailu and G. Sommer. On amount and quality of bias in reinforcement learning. In *International Conference on Systems, Man, and Cybernetics (SMC)*, 1999. 20, 24

[71] S. Hansen, W. Dabney, A. Barreto, T. Van de Wiele, D. Warde-Farley, and V. Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations (ICLR)*, 2020. 20

[72] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 36, 38, 50

[73] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition*, pages 9729–9738. IEEE, 2020. 50

[74] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 19

[75] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel. Curiosity-driven exploration in deep reinforcement learning via bayesian neural networks. *CoRR*, abs/1605.09674, 2016. 5

[76] J. M. Hunt. Intrinsic motivation and its role in psychological development. In *Nebraska Symposium on Motivation*, volume 13. University of Nebraska Press, 1965. 5

[77] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[78] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 11(Apr):1563–1600, 2010. 19

[79] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 19

[80] L. Ke, J. Wang, T. Bhattacharjee, B. Boots, and S. Srinivasa. Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. In *International Conference on Robotics and Automation*. IEEE, 2021. 50

[81] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002. 19

[82] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2017. 8

[83] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, 114(13):3521–3526, 2017. 19, 20

[84] A. S. Klyubin, D. Polani, and C. L. Nehaniv. All else being equal be empowered. In *European Conference on Artificial Life*, 2005. 19

[85] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021. 51

[86] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski. The darpa robotics challenge finals: Results and perspectives. *Journal of Field Robotics*, 34(2):229–240, 2017. 43, 45

[87] A. Kumar, T. Buckley, J. B. Lanier, Q. Wang, A. Kavelaars, and I. Kuzovkin. Offworld gym: open-access physical robotics environment for real-world reinforcement learning benchmark and research. *arXiv preprint arXiv:1910.08639*, 2019. 45

[88] A. Kumar, A. Singh, F. Ebert, Y. Yang, C. Finn, and S. Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022. 32

[89] V. Kumar and E. Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *International Conference on Humanoid Robots*, pages 657–663. IEEE, 2015. 48

[90] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22, Mar 1985. 19

[91] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 33

[92] J. Lau, B. Zimmerman, and F. Schaub. Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW), 2018. 16

[93] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019. 32, 34

[94] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. *2019 International Conference on Robotics and Automation (ICRA)*, 2019. 6

[95] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. 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, 2018. 45

[96] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. *arXiv preprint arXiv:2212.03858*, 2022. 32, 33, 36

[97] W. Li, J. Konstantinova, Y. Noh, Z. Ma, A. Alomainy, and K. Althoefer. An elastomer-based flexible optical force and tactile sensor. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 361–366. IEEE, 2019. 33

[98] Z. Liu, W. Liu, Y. Qin, F. Xiang, M. Gou, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun, et al. Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation. *IEEE Robotics and Automation Letters*, 7(1):486–493, 2021. 45

[99] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 32, 34

[100] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023. 32, 34

[101] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018. 44, 45

[102] E. Mansimov and K. Cho. Simple nearest neighbor policy method for continuous control tasks, 2018. URL https://openreview.net/forum?id=ByL48G-AW. 51

[103] P. Morgado, I. Misra, and N. Vasconcelos. Robust audio-visual instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12934–12945, 2021. 34

[104] P. Morgado, N. Vasconcelos, and I. Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021. 33, 34, 35, 37

[105] B. A. Morrongiello, K. D. Fenwick, and G. Chance. Crossmodal learning in newborn infants: Inferences about properties of auditory-visual events. *Infant Behavior and Development*, 21(4), 1998. 5

[106] A. Murali, Y. Li, D. Gandhi, and A. Gupta. Learning to grasp without seeing. In *International Symposium on Experimental Robotics*, pages 375–386. Springer, 2018. 32, 34

[107] A. Murali, Y. Li, D. Gandhi, and A. Gupta. Learning to grasp without seeing. *International Symposium on Experimental Robotics*, 2018. 6

[108] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 32, 34, 36, 38, 50

[109] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020. 19

[110] A. Nichol, V. Pfau, C. Hesse, O. Klimov, and J. Schulman. Gotta learn fast: A new benchmark for generalization in rl. *arXiv preprint arXiv:1804.03720*, 2018. 10

[111] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 38, 40

[112] I. Osband, B. V. Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research (JMLR)*, 20(124):1–62, 2019. 19

[113] G. Ostrovski, M. G. Bellemare, A. van den Oord, and R. Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning (ICML)*, 2017. 17

[114] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11 (2), 2007. 7

[115] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6, 8, 9

[116] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 34, 51

[117] S. Parisi, V. Dean, D. Pathak, and A. Gupta. Interesting object, curious agent: Learning task-agnostic exploration. *Advances in Neural Information Processing Systems*, 34: 20516–20530, 2021. 2

[118] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. K. Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *ICML*, 2022. 50

[119] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017. 4, 5, 6

[120] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, 2017. 17, 19, 20, 22, 23, 25

[121] D. Pathak, D. Gandhi, and A. Gupta. Self-Supervised exploration via disagreement. In *International Conference on Machine Learning (ICML)*, 2019. 4, 5, 11, 12

[122] M. Patrick, Y. M. Asano, P. Kuznetsova, R. Fong, J. F. Henriques, G. Zweig, and A. Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020. 34

[123] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. The robotarium: A remotely accessible swarm robotics research testbed. In *International Conference on Robotics and Automation*, pages 1699–1706. IEEE, 2017. 45

[124] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *International Conference on Robotics and Automation*, pages 3406–3413. IEEE, 2016. 45

[125] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 50

[126] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 34

[127] R. Raileanu and T. Rocktäschel. RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments. In *International Conference on Learning Representations (ICLR)*, 2020. 19, 22, 23, 25

[128] J. Rajendran, R. Lewis, V. Veeriah, H. Lee, and S. Singh. How should an agent practice? In *Conference on Artificial Intelligence (AAAI)*, 2020. 21

[129] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning (ICML)*, 2019. 19, 20

[130] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, 2015. 17, 19

[131] M. B. Ring. *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin Austin, Texas 78712, 1994. 20

[132] M. B. Ring. CHILD: A first step towards continual learning. In *Learning to learn*, pages 261–292. Springer, 1998. 20

[133] P. Rochat. Object manipulation and exploration in 2-to 5-month-old infants. *Developmental Psychology*, 1989. 1

[134] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 20

[135] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, et al. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019. 31

[136] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell. Policy distillation. In *International Conference on Learning Representations (ICLR)*, 2015. 20

[137] R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000. 19

[138] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *International Conference on Computer Vision (ICCV)*, 2019. 5, 10, 25, 29

[139] J. Schmidhuber. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks*, 1991. 5, 7

[140] J. Schmidhuber. A possibility for lmplementing curiosity and boredom in Model-Building neural controllers. In *International Conference on Simulation of Adaptive Behavior (SAB)*, 1991. 5, 17, 19

[141] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006. 19

[142] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 10

[143] M. Schultheis, B. Belousov, H. Abdulsamad, and J. Peters. Receding horizon curiosity. In *Conference on Robot Learning (CoRL)*, 2019. 19

[144] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine learning (ICML)*, 2018. 20

[145] G. Seetharaman, A. Lakhotia, and E. P. Blasch. Unmanned vehicles come of age: The darpa grand challenge. *Computer*, 39(12):26–29, 2006. 43, 45

[146] T. Seno and M. Imai. d3rlpy: An offline deep reinforcement learning library. *arXiv preprint arXiv:2111.03788*, 2021. 51

[147] P. Sharma, L. Mohan, L. Pinto, and A. Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on Robot Learning*, pages 906–915. PMLR, 2018. 45

[148] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. Loss is its own reward: Self-supervision for reinforcement learning. In *International Conference on Learning Representations*, 2017. 5

[149] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, 2012. 6

[150] B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *CoRR*, abs/1507.00814, 2015. 5

[151] B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. In *NIPS Workshop on Deep Reinforcement Learning*, 2015. 19

[152] S. Still and D. Precup. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3), 2012. 5

[153] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 10, 14, 29

[154] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences (JCSS)*, 74(8): 1309–1331, 2008. 19

[155] Y. Sun, J. Falco, M. A. Roa, and B. Calli. Research challenges and progress in robotic grasping and manipulation competitions. *IEEE Robotics and Automation Letters*, 7(2): 874–881, 2021. 45

[156] B. Sundaralingam, A. S. Lambert, A. Handa, B. Boots, T. Hermans, S. Birchfield, N. Ratliff, and D. Fox. Robust learning of tactile force estimation through robot interaction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9035–9042. IEEE, 2019. 33

[157] A. A. Taïga, W. Fedus, M. C. Machado, A. Courville, and M. G. Bellemare. Benchmarking bonus-based exploration methods on the arcade learning environment. In *International Conference on Learning Representations*, 2020. 13

[158] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 5, 29

[159] Y. W. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu. Distral: Robust multitask reinforcement learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 20

[160] A. Thankaraj and L. Pinto. That sounds right: Auditory self-supervision for dynamic robot manipulation. *arXiv preprint arXiv:2210.01116*, 2022. 32, 33, 34

[161] S. Thrun and T. M. Mitchell. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1-2):25–46, 1995. 20

[162] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems (IROS)*, 2012. 44

[163] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 43

[164] K. Watanabe and S. Shimojo. When sound affects vision: effects of auditory grouping on visual motion perception. *Psychological Science*, 12(2), 2001. 5

[165] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 42

[166] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016. 19

[167] A. White, J. Modayil, and R. S. Sutton. Surprise and curiosity for big data robotics. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 7

[168] T. Wilcox, R. Woods, C. Chapa, and S. McCurry. Multisensory exploration and object individuation in infancy. *Developmental psychology*, 43(2), 2007. 5

[169] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning (ICML)*, 2021. 20

[170] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020. 44

[171] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 33

[172] H. Zhao, C. Gan, W.-C. Ma, and A. Torralba. The sound of motions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1735–1744, 2019. 34

[173] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, et al. Train offline, test online: A real robot learning benchmark. *arXiv preprint arXiv:2306.00942*, 2023. 36, 37

[174] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning, 2020. 44