ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Semester Thesis

Worldloop Transfer

Evaluating perceptual capabilities learned through embodied tasks

Autumn Term 2022

Supervised by:
Dr. Cesar Cadena
Prof. Amir Zamir (EPFL)

Author:
Yi Fei (Jessica) Bo

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

WORLDLOOP TRANSFER

Authored by (in block letters):

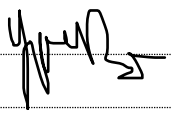
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Bo	Yi Fei

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Cambridge MA, 28.02.2022	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

Abstract

How do the perceptual abilities developed by embodied agents correlate to the active behaviours they execute in the 3D world? Parallels between computational actors, namely reinforcement learning (RL) agents, can be drawn with ecological agents through the principle of the action-perception-loop (APL), which describes the interlinked relationship between learning to *see* and learning to *act*.

We investigate how the APL manifests in RL agents trained on a variety of embodied tasks with different perceptual requirements, including motor movements, visually-guided navigation, and localization. In the absence of a holistic perceptual evaluation method, we use depth estimation as a substitute task to evaluate the agent's learned visual encoder post-training. Our preliminary results find that agents with smoother policies encode significantly better depth perception abilities than a scratch encoder, but performance differences between task variations are difficult to quantify given inconsistencies in training agents to convergence.

As a further step, we explore if APL can be applied to address technical challenges in RL, such as sample efficiency in training. We initialize agents with pretrained depth visual encoders and repeat their training to compare with the scratch agents. We find that some embodied tasks which require visual information saw benefits in the jumpstart reward performance, but definitive results are limited by agents overfitting to the training environments. We make a list of recommended changes to the technical setup of the experiments and suggest future directions.

Chapter 1

Introduction

The action-perception loop (APL) describes the inherent relationship between *seeing* and *doing* in an environment. In J. J. Gibson's [1] theory of ecological psychology, embodiment is at the forefront of visual cognition. Perception of the environment equates to learning the affordances of the objects present, the understanding of which then guides the actions that a cognisant agent can execute. In the adjacent fields of cognitive science and neuroscience, recent research directions focus on approaching the motor and sensory functions of the brain as integrated units with significant influence over one another, rather than as separate pathways [2]. Even in evolutionary biology, the development of visual systems is shown to match the visually-guided behaviours of animals. In a positive feedback loop, the capability to process visual information develops in correspondance to the requirements of the ecological behaviour of the animal [3].

While the scientific understanding of the APL in natural science fields is not absolutely unified across all disciplines, there is no question that the high-level principles about its relationship hold true. Thus, it is valuable to also apply an interconnected model of vision and behaviour towards computational models of embodied intelligence. In the field of robotics, APL manifests through active vision, which describes a goal-oriented strategy for data acquisition through active behaviour, such as moving through the environment and focusing attention on areas of interest [4]. The visual abilities of the robotic agent must be supportive of extracting appropriate and useful information from the environment to select actions that fulfill some overarching objective. Such a compatibility between the perception and action is interesting to study, as it suggests there requires a level of sufficiency for a visual system to meet the needs of an embodied task, and achieving such a system can benefit developing more complex behaviours in the agent.

For the parameters of this project, we focus on intelligent agents that capable of action through reinforcement learning (RL) policies. Given environmental observations as input, the agent selects the best action within its action space to maximize the reward it receives. Pervasive problems in reinforcement learning include instability and sample efficiency, especially when it comes to optimizing policies for highly complex navigation tasks [5]. Therefore, the understanding of what perceptual abilities arise from training an embodied task, and correspondingly what perceptual information are beneficial for training an embodied task, can both alleviate practical concerns of RL training and investigate behaviour-driven perception.

We propose to investigate both directions of the APL through a two-staged experiment. We first isolate the contribution of action on perceptual development, then explore the contribution of perception on learning active behaviours. **Experiment A** evaluates the perceptual abilities that an active RL agent gains as the result of the embodied task that it is trained to perform. We theorize that through modulating the task parameters, the visual system of the agent would develop in accordance with the task parameters. **Experiment B** repeats the embodied task training with agents that are initialized with existing perceptual abilities. We hypothesize that the embodied tasks which require such perceptual skills would benefit the most in terms of training speed (sample efficiency) or generalization to unseen environments.

Executing these experiments require determining an appropriate perceptual metric and a set of embodied tasks with controlled variations in the task parameters. Neither objective is well-defined in standard computer vision and reinforcement learning research. There does not exist any measures of embodied visual intelligence (even for humans), so this project uses depth perception as a proxy metric. Depth is relevant to action for several reasons, including its connection to the development of independent movement [6] and the visual anatomy of animals eyes to support their depth perception abilities based on their active behaviours [3]. Depth information can be easily captured through RL training simulators, thus minimizing the technical complexities of the implementation. Chapter 3 introduces the full motivation for using depth as the perceptual task and provides details of the implementation of the metric.

For the embodied task variations, the main goal is to probe which task parameters induces changes in agent's learnt perceptual abilities. The implementation of the agent follows standard RL conventions and are described in Chapter 4. We outline a set of RL training task objectives that increments in complexity, from random movements, to motor-based skills like 'move forward', to more complex navigation-based behaviours like 'move to *this location*'. For our purposes, an embodied task is defined according to a high-level task objective through a reward function that encapsulates several reward axes, each one representing an embodied skill that is interpretable in the biological context. For example, a negative penalty for collisions would encourage 'collision-avoidant' behaviour. A set of intra-task and inter-task reward variations are explored to determine how they contribute to the development of active depth perception. The experiment details are provided and justified in Chapter 5.

The bridge between active task training and static perceptual evaluation lies in the capability to agilely isolate the perceptual network of the agent. The advantage of computational model over human test subjects is the ability to directly access the encodings of the 'visual cortex' for downstream experiments. We apply transfer learning to the extracted the visual encoder from the trained agent as the evaluation method for depth perception. This technique is widely applied to transferring knowledge between overlapping task domains [7]. Humans also demonstrate that training in one visual task results in an improvement in transferred performance in a related visual task [8, 9]. On the neurological level, it is hypothesized that neurons undergo reweighting to fine-tune to the new task, drawing parallels to weight updates in a neural network [10]. The technical implementation is described in Chapter 4.

In summary, this project investigates if depth perception (as a proxy of general perceptual intelligence) arises from training an RL agent on embodied tasks of various objectives and skill axes. Furthermore, we compare and contrast if the

presence of depth perception abilities improves the outcomes of training a given embodied task. Establishing a bidirectional relationship where action influences perception, and vice versa, would demonstrate the existence of a computational version of the APL. The formulation of these experiments can be extended in the future to other perceptual task domains as well. In the best case scenario, depth estimation would be replaced with an idealized metric that is able to fully evaluate intelligent visual systems.

Chapter 2

Related Works

This section explores the related literature across a range of fields. Since we are not building directly off of any specific studies, the literature search is not restricted to the technical domain, but attempts to identify works that bridge the gap between artificial and natural intelligence. We review works pertaining to offline computer vision, active computer vision, robotics, reinforcement learning, and transfer learning. Although there is a strong history of neuro-inspired robotic systems, most works were completed prior to the popularization of deep learning methods and thus are not considered high-impact in the current research landscape. More recently, there has been increased interest within the machine learning community regarding techniques to improve reinforcement learning performance using methods such as transfer learning, representation learning, and self-supervised learning.

In general, we do not find any works that use deep reinforcement learning to directly study the full action-perception loop in embodied agents. The closest aligning work is [11], which studies how interactive RL agents trained with gameplay develop visual representations of the world. The agents are evaluated on both static image datasets (similar to us, but with a wider variety of tasks including depth, normals, classification, and affordances) and dynamic tasks that require the integration of temporal information, which are inspired by classical psychology tasks.

Computer Vision: Classic computer vision (CV) typically approaches visual evaluation with standalone tasks on large static image datasets. Tasks such as depth estimation [12, 13], semantic segmentation [14], object classification [15] have received incredible amounts of research attention and are associated with standardized performance benchmarks. The goal of this project is not to improve on the state of art classification or prediction performance in offline visual tasks, but approach visual learning as an action-driven process. We use depth estimation as the perceptual metric to evaluate active agents, but the emphasis is on the relative performance between agents, not the absolute performance (see Chapter 3 for our metrics).

Active Vision: In simple terms, active vision is differentiated from offline computer vision with the assumption that the observer is active in the environment and able to control sensing parameters [16]. It can be framed as a strategic data sampling for active tasks, as such selecting where in the environment to observe to maximize the gain of information [4, 17]. Complex visual tasks see benefits when approached as an active visual task rather than a passive visual task (at least prior to deep learning) [18, 19]. There are also significant applications in robotics, in-

cluding navigation and localization. Overall, technical works in active vision mostly focus on sensor planning [20], where we intend to focus on the idea of embodiment. This is similar to the much more underdeveloped field of enactive vision, which describes cognition and perception from a sensorimotor framework [21, 22].

Bio-Inspired Robotics: Predating the advancement of deep learning-driven reinforcement learning (DRL), neuroscience and psychology have been used to guide the design of to robotic systems and controls, including the visual processing module. In such works, roboticists take inspiration from neuronal organization or behavioural-driven abilities of insects and animals to handcraft similar features in robots [23, 24, 17]. This method is not common anymore, as it is not reasonable to expect that a computational system, under its own set of unique constraints and task specifications, would exactly reflect the composition and behaviour of natural intelligent systems. We do not use nature to guide the design of our agent, but rather we investigate if the APL principle can be observed beyond natural domains and leveraged for the advantage of RL training.

Reinforcement Learning: A wide discipline [25] with many applications from games [26] to manipulation [27] to navigation [28], RL solves a class of problems that requires making decisions on acting in an environment. Rather than learning in a supervised manner on labelled data, RL agents are trained on a reward signal to develop a policy for acting that maximizes the said reward. While powerful, there are many problems with associated with this paradigm of learning, including the exploration vs. exploitation trade-off [29], reward formulation, generalizability, and sample efficiency [5]. The latter is particularly impactful to pixel-based DRL in complex environments, as agents struggle with extracting useful state representations from rich image inputs. Works have targeted sample efficiency problems with (a non-definitive list of) methods like meta-learning [30], representation learning [31, 32, 33], self-supervised learning [34, 35], data augmentation [36], and expert demonstrations [37, 38].

Transfer Learning: When dealing with limited training samples, transfer learning (TL) can be used to share knowledge representations between similar but discrepant tasks [7, 39], including in visual domains [40]. In the RL context, TL can be leveraged in such ways as directly transferring between RL tasks [41], or pretraining the agent with useful state representations [42, 43, 44, 45]. Several works find that the representations learned through source tasks or exploratory tasks improves the results in the downstream task training [46, 47].

Most relevant to our methods is [48], which investigates how visual priors can be used to enhance the training of visuomotor navigation and localization agents, and [49] which does the same for manipulation agents. Both find that pretraining relevant visual tasks result in better generalizability and learning speed for the agent. We take the approach of transferring visual encoders trained on depth estimation to the agent, which can be regarded as a limited task subset of [48]. In an ideal scenario, we would be pretraining on a set of tasks that are optimal for embodiment.

Chapter 3

Perceptual Metrics

Static image tasks on large RGB image datasets (e.g. classification on ImageNet [50]) are often treated as the ubiquitous way to evaluate vision networks, but this unveils little about perception necessitated by action. If we consider a biological model of visual intelligence, there exists a number of tasks used by cognitive scientists and psychologists to assess visual skill. These range from low-level (confined to a basic perceptual dimension) like orientation discrimination, to high-level (related to identifying natural objects) like face recognition [10]. However, the same problem arises that the acquiring of these skills are not necessarily driven by action, thus we must turn to another source. As mentioned in Chapter 1, the proxy perceptual skill selected is depth estimation, which will be explained in more detail here.

3.1 Depth Perception

The infant's gradual acquisition of depth perception is a topic of interest in developmental psychology [51]. It has been observed that depth is likely linked to active tasks like eye and head movements, reaching [51], and locomotion [6]. The ability to perceive depth is not enabled by one singular function, but rather arises from a combination of visual cues that contribute varying depending on the parameters of the scene. Binocular vision, which utilizes the angle of which the eyes focus at, plays a larger role in close-by settings, where the sensitivities to convergence is higher [52]. Monocular people have reduced depth perception, but can rely on cues like motion parallax, perspective, and familiar size of objects to judge distances [53]. Figure 3.1 from [54] depicts the taxonomy of depth cues.

The *visual cliff* experiments [6] depicted in Figure 3.2 aimed to investigate depth perception through evaluating responses of locamotive babies and animals placed near a glass drop-off. The study identified that some form of depth understanding is present during the development of independent movement. This follows the principles of ecological psychology, which believes that perception is for understanding affordances of the environment. While their conclusion is broad and does not examine the different contributing depth cues in isolation (as it is actually the case that some types of depth perception are present from birth [51]), depth understanding presents us with a valuable and interesting perceptual skill to investigate.

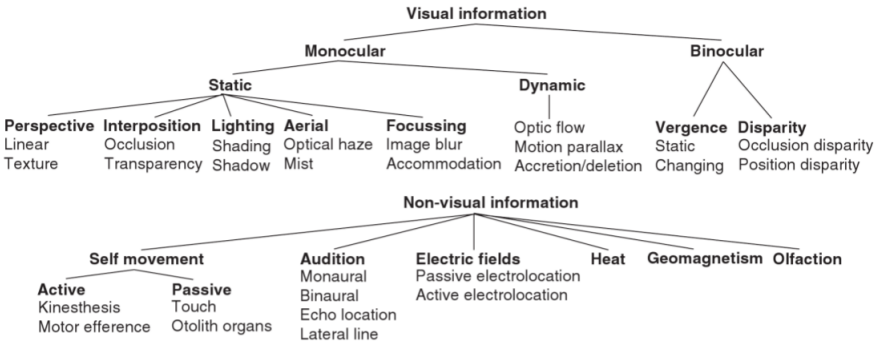


Figure 3.1: Taxonomy of depth cues from [54].

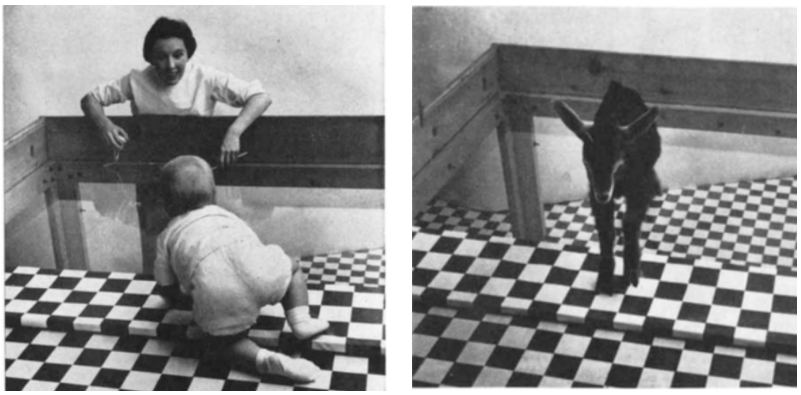


Figure 3.2: Visual cliff experiments where an infant and young goat respectively are placed on glass platforms to simulate a cliff [6].

Animals' depth perception abilities varies depending on their ecological role and is enabled by their visual anatomy, see Figure 3.3. Ambush predators have elongated pupils to reduce blur in their depth of field to better estimate distances, while prey animals do not require this skill, thus giving them horizontally expanded pupils for panoramic vision [3]. This gives rise to the implication that if robots in the wild can curate their own visual system from active tasks, they should develop different depth sensitivities based on the parameters of their task and the limitations of their onboard sensors. In our specific application, we consider a monocular RGB camera as the only source of input information to the agent. This creates some interesting restrictions as any cues from binocular vision and other senses are eliminated. However, even with a single RGB camera, depth can still be signalled through a variety of ways — optical flow, familiar sizes, texture deletion, and occlusion.

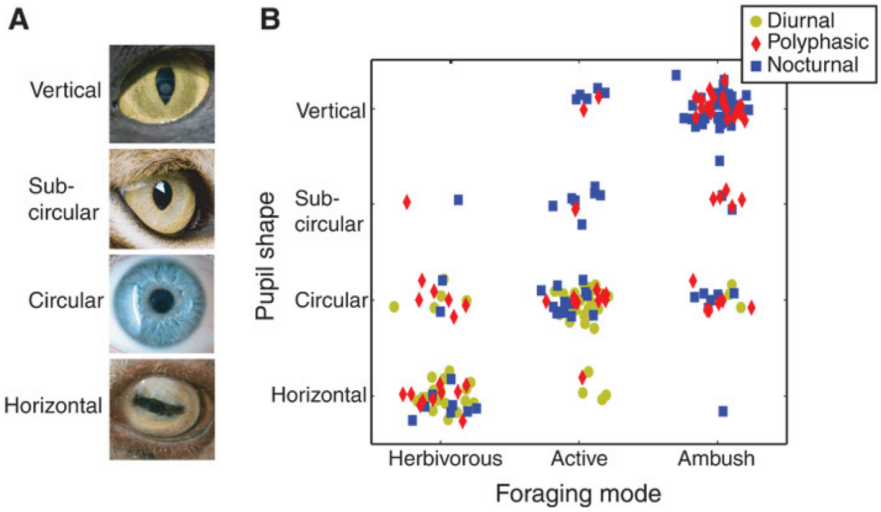


Figure 3.3: Animal pupils and distribution of their ecological foraging modes, adapted from [3]

3.2 Depth Dataset

Despite the sensory limitations of the computational model, there is still benefit to investigate depth as a perceptual task. A great advantage is that depth ground truth is readily available via the Gibson dataset of scanned 3D environments [1], which the agent trains within via the Habitat simulator [55]. Gibson contains 572 buildings composed of 1,447 floors covering a total area of 211,000 m^2, but only 90 compatible buildings are used in this project. While other image datasets like Taskonomy [40] contain a wider variety of mid-level visual tasks, they were not considered at this stage due to the potentially significant domain gap between the active training and transfer learning stages.

We capture datasets of RGB images and corresponding depth maps by sampling 500 random viewpoints and locations from navigable areas of each environment. This can be done by attaching RGB and depth sensors (both of image resolution 128x128) to the simulation environment and saving the outputs of the observation. Any areas where depth is not available, such as outdoor spaces or glass, the pixels are masked out and not included in the loss computation. The range of depth varies

approximately between 0-14 meters to the image plane. The training and evaluation dataset splits are detailed in Chapter 4.

3.3 Non-linear Depth Experiments

Although pixelwise depth estimation was chosen as the measurable task, we initially considered that it would not be optimal for capturing active vision skills. To given an example, humans are not very good at predicting raw distance values as we overestimate close distances and underestimate far distances [56]. We experimented with capturing non-linearities in depth perception by applying masks to the depth maps. In this section, we evaluate a trained agent on these depth estimation task variations via our transfer learning setup. The technical implementations of the setup are detailed later in 4. For a given depth variation task, we see if a pretrained RL agent can outperform a scratch agent to a *higher degree* in terms of evaluation loss on the task in comparison to vanilla depth estimation. If shown to be true, then the non-linear variation of depth is a better match to the depth representation that the agent encodes.

We trialled three depth task variations – saliency-weighted mask, distance-based mask, and semantic-based mask on the class 'floor'. It was eventually found that none of these methods performed significantly well to be reasonably justified, so we proceeded with vanilla depth estimation in the experiments. The outcomes of the weighted depth tasks are reported as follows.

3.3.1 Saliency-Weighted Depth

Attention is known to be an extremely important mechanism in active vision to select the most relevant information for visual processing [57]. In computational terms, the model's saliency map shows the areas of the input image that it focuses on to solve its training task. We hypothesized that an active agent should be better at processing areas of the input image that are relevant for movement or navigation. Thus, weighting the pixels of the depth map with the agent's saliency map may result in better performance in the depth estimation task.

Saliency is extracted from a pretrained active agent by averaging the gradients given an input image. The saliency map is computed for every image in both the training and testing sets, then applied to the loss function as a weight for each pixel during training [1]. Figure 3.4 shows sample model saliency maps from navigation agents trained on different tasks. The first two RGB-input agents (*PointNav* and *ForwardNav*) are trained by us. The specifics of these task objectives are described in Chapter 6 as their implementation is not highly relevant here. The last column is an RGBD-input *ObjectNav* agent trained and provided by the Habitat team at Facebook AI Research. We include it as reference since it is good at identifying navigable spaces and target zones, so its saliency map is considered to be more optimal for navigation using depth information. In contrast, the two RGB models seem to fixate randomly on areas that are not interpretable, so it is unclear if their saliencies are actually 'good'.

Figure 3.5 shows a comparison of the training loss of the saliency-weighted depth

[1] The saliency map computation and weighting was implemented by another student.

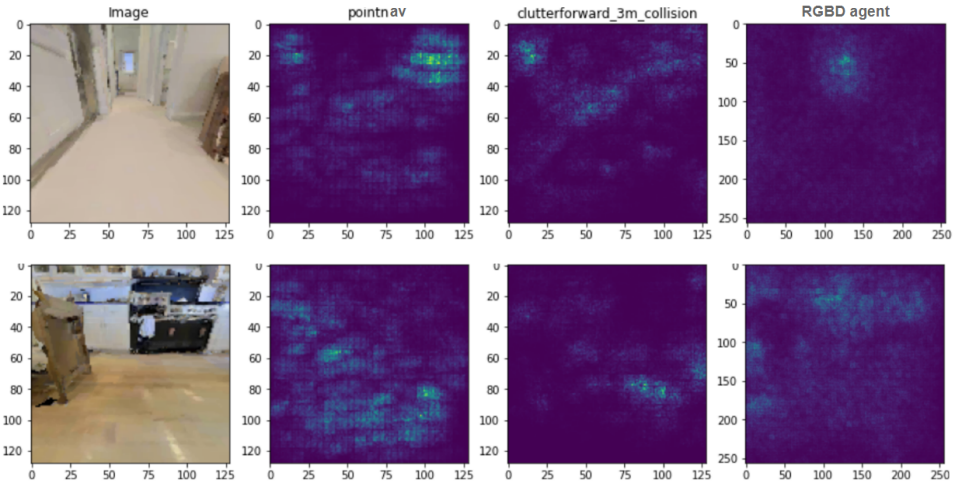
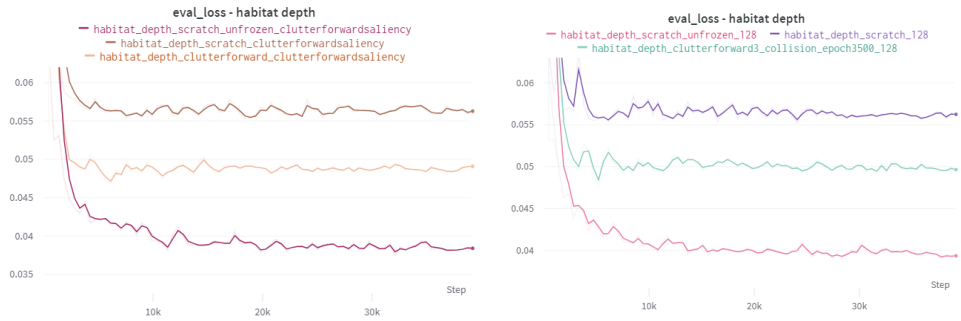


Figure 3.4: Saliency maps from three models: RGB *PointNav* agent, RGB *ForwardNav* agent, and RBGD *ObjectNav* agent.

(left) and the vanilla depth (right). This analysis was done informally with the intention of doing a visual review of the results. In each figure, the top run is a frozen scratch agent, the middle run is the frozen pretrained *ForwardNav* agent, and the bottom run is the unfrozen scratch agent. The same pretrained agent was used in both trials to showcase its relative performance in comparison to frozen and unfrozen scratch encoders. Surprisingly, we do not observe a significant improvement in the agent's relative performance. This was repeated for agents trained on other embodied tasks (including the RGBD agent) and the same result persisted. Due to this outcome, we decided to not pursue saliency weighting.



(a) Depth estimation with pixels < 4m.

(b) Vanilla depth estimation.

Figure 3.5: Comparison of distance-weighted and vanilla depth estimation

3.3.2 Distance-Masked Depth

Inspired by natural non-linearities in human vision where depth sensitivity is decreased at far distances due lowered resolution of depth cues like convergence and motion parallax, we mask out all pixels more than 4 meters away from the image plane. This is done simply by not including these pixels in the loss computation.

Figure 3.7 shows a comparison of the evaluation loss on the distance-masked depth

and vanilla depth tasks. While there is a slight observable improvement with distance-masking, we did not deem it to be significant enough to justify switching to. The experiment repeated on masking out pixels over 2 meters away showed visually similar results. Perhaps distance-masking (or a more sophisticated form of distance-weighting) can be incorporated into a future perceptual metric, but we decide to disregard it for now.

(a) Depth estimation with pixels < 4m.

(b) Vanilla depth estimation.

Figure 3.6: Comparison of distance-weighted and vanilla depth estimation.

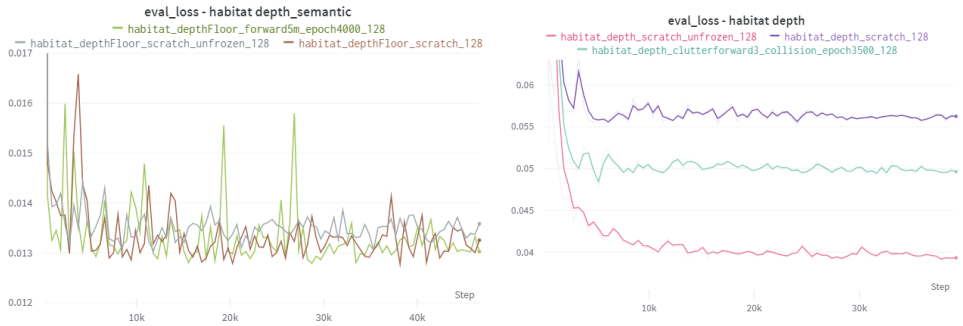
3.3.3 Semantic-Masked Depth

Similar to the idea of saliency-weighting, here we focus on a particular semantic class, the 'floor'. The floor plane is interesting in two parts. 1) It represents the navigable spaces, which should be important for embodied agents. 2) The depth of the floor is very predictable if perspective is understood. Since none of the floors in the environment are known to have slopes, then all of the 'floor' pixels at the same height in an image would have the same depth, making it a computationally simple problem.

Due to the reason that the problem is easy to optimize, we find that resulting losses have very low values, and are consequentially highly noisy. Through examining the predicted depth maps, it is apparent that the transfer learning model converged to a local minimum, where it predicts the same depth gradient for each image. This fulfills the loss objective perfectly and there's no advantages to having a good visual encoder. Thus, we deem that the investigation was ill-posed for the given semantic class. A possible idea to try in the future is estimating distances of collider objects, but this is harder to build a dataset from.

3.4 Other Perceptual Metrics

In this project, for the ease of implementation, we focus on pixelwise vanilla depth estimation. However, it is undeniable that intelligent perceptual understanding is much richer and complex, involving higher order cognitive functions and reasoning abilities. For example, [11] shows that embodied RL agents trained in interactive environments learn representations of object permanence, free space, and containment. Other candidates for tasks can be semantic scene understanding [58], affordance reasoning [59], and violation-of-expectation prediction [60]. These skills are



(a) Depth estimation with floor pixel.

(b) Vanilla depth estimation.

Figure 3.7: Comparison of saliency-masked and vanilla depth estimation.

apparent in human abilities, but such datasets would result in too large of a domain gap to be used in this project context.

In parallel to this work, we are also investigating automated methods for mining metrics/losses that better captures the visual non-linearities that embodied agents learn, instead of testing out each one manually as the previous section does [2]. In addition to depth estimation, we would also expand to include other perceptual tasks, forming a set of skills that represents the basic dimensions of perceptual intelligence, upon on which any visual system can be evaluated. With a complete set of perceptual abilities, we can holistically evaluate different embodied tasks trained agents with more fine-grained nuances and draw more precise conclusions. But of course, this is all very theoretical at the moment.

[2]The metrics mining project is conducted by another student.

Chapter 4

Technical Implementation

This chapter describes the technical implementation of the RL agent and the transfer evaluation learning framework. The merit of the experimental results hinges on the implementation decisions of the APL, which is of course not without some limitations. However, emulating biological or ecological concepts computationally has always been a challenge that requires a number of tradeoffs to preserve implementability, complexity, and time and resource requirements. With that being said, the focus of this project is not in selecting the best technical design [1], but rather the experimental design, which is covered in Chapter 5.

4.1 RL Agent

The recent resurgence of reinforcement learning parameterized by deep neural networks (DRL) provides a powerful and flexible technical platform to support active learning agents [61]. Previous high-dimensional intractable problems can now be solved with relative ease, sometimes even improving upon human performance [62]. DRL is also no doubt the method of choice for problems centered around learning action policies, which gives our work practical implications for being incorporated into future research. As in typical RL problem setups, we employ an agent that observes the current state s_t at time t, selects an action a_t based on its learned policy, and then transitions to the next state s_{t+1} and receives a reward r_{t+1}. The reward function that determines r_t for each state and action formulates the embodied task objective (which is fully detailed in Chapter 5).

4.1.1 RL Architecture

The agent itself does not receive raw inputs of the state, which are 128x128x3 RGB images of the environment, but rather state representations generated through passing the inputs into a visual encoder (ResNet-18 [63]) and a single-layer gate recurrent unit (GRU). The GRU is a mechanism which allows the propagation of previous states into the current state processing, which enables memory of previous timesteps. The ResNet-18 is initialized with random values in **Experiment A**.

[1] The major design choices of using RL and transfer learning were made before starting.

For **Experiment B**, it is replaced by a ResNet-18 encoder pretrained on depth estimation using a large baseline dataset sampled in the same way as the transfer learning dataset from Section 4.2.

The output action can be one of three choices from: {*turn left 30°*, *turn right 30°*, *go forward 0.25 meter*}. The restrictive action space does represent one of the major limitations. There is no fine grained motor control, height variations, speed modulation, sensor tuning, or even physical interactions with the environment due to the constraints of the simulator. Upon the executing the action, the agent's position in the environment is updated and the input images are reflective of the new state.

We implement the RL agent with a proximal policy optimization (PPO) algorithm [64] in an actor-critic architecture [2]. To summarize its advantages, PPO restricts the size of the policy update through clipping a surrogate policy gradient objective function, resulting in better stability and sample-efficiency for training. Otherwise, the high variance of training can cause the policy updates to be too large and venture outside of the trust region and destroy the policy altogether. The overview of the DRL components are presented in Figure 4.1. All work was done using the PyTorch library [65]. The hyperparameters were kept consistent since the objective is not to achieve competitive performance on training tasks, but to hold all non-task related parameters constant.

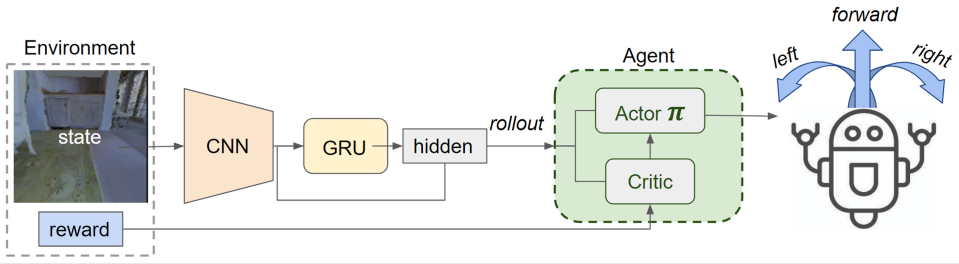


Figure 4.1: RL agent architecture.

4.1.2 Training and Environments

For the training environment, we use the Habitat simulator [55] with the Gibson dataset of indoor environments, which captures real-world 3D spaces with varying levels of navigation complexity [1]. Habitat-Sim is noted to be one of the only existing simulators that supports realism, however it does lack advanced physics and interaction abilities [66]. Since we are concerned with navigation-based tasks, interactivity is less important in this scope.

Due to computational limits, only a maximum of 15 environments can be run in parallel for training the agent, which we divide into 10 fixed training environments and 5 fixed evaluation environments. The 10/5 training-evaluation split was used for a majority of the task training, with the exception of highly complex tasks that benefited from more diverse training environments, in which case the evaluation environments were moved to the training set for a 15/0 split.

Each training episode consists of 1000 steps. During updates, the agent samples from full executed trajectories, which is necessary due to the recurrent aspect of the

[2]The base DRL architecture and training scripts were implemented by another student prior.

policy that introduces temporal reliance on previous state transitions in an episode. Training was manually terminated when it was determined to reached convergence through its accumulated rewards (either the rewards have reached an optimal value or they have stopped improving for a significant period of time despite being suboptimal). This can occur between as little 500 or as many as 8000 training iterations, depending on the complexity of the training task. We also continued many training runs past the point of convergence to observe post-convergence perceptual development.

4.2 Transfer Learning Framework

In Chapter 3, we compared several weighted variations of the downstream task and ultimately selected vanilla depth estimation, which is the per-pixel metric depth from the image plane. This section describes how the visual encoder from the trained agent is ported into a static setting to predict pixel-wise depth.

4.2.1 Transfer Learning Architecture

The objective of transfer learning is to take the learned visual representations from the embodied task and evaluate them on a downstream visual task [3]. Since our depth prediction task is pixel-to-pixel, we must upsample the encoded representations back to an image of the original resolution. We use a modified U-Net architecture [67], which involves downsampling (encoding) followed by upsampling (decoding) an image to perform predictions. In our case, we only require the decoder since the ResNet-18 from the agent's visual module acts as the encoder.

The original architecture of U-Net introduces a critical feature of concatenating the downsampled feature maps to the corresponding upsampling steps, which serves to preserve the resolution of the original image. However, since we want to restrict the decoder layers to only access the final encoded representations of the image, these skip connections were removed. In theory, this makes the architecture similar to a fully convolution network (FCN-32) without skip connections [68]. As shown in Figure 4.2, we map the pretrained agent's image encoder output to a U-Net decoder with five levels of upsampling. This causes the output depth maps to lose detailed resolution, but it is satisfactory for our use case since we aim to isolate and compare performance between different encoders. Figure 4.3 shows the difference in predictions of our transfer learning setup with and without skip connections.

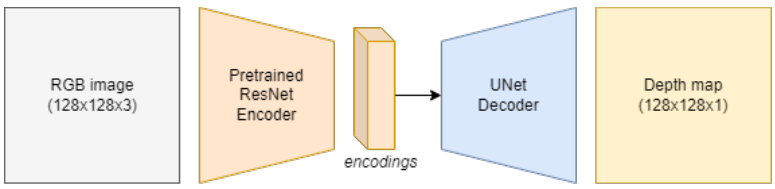


Figure 4.2: Our transfer learning framework.

[3]The transfer learning pipeline was implemented prior to the start of the semester project.

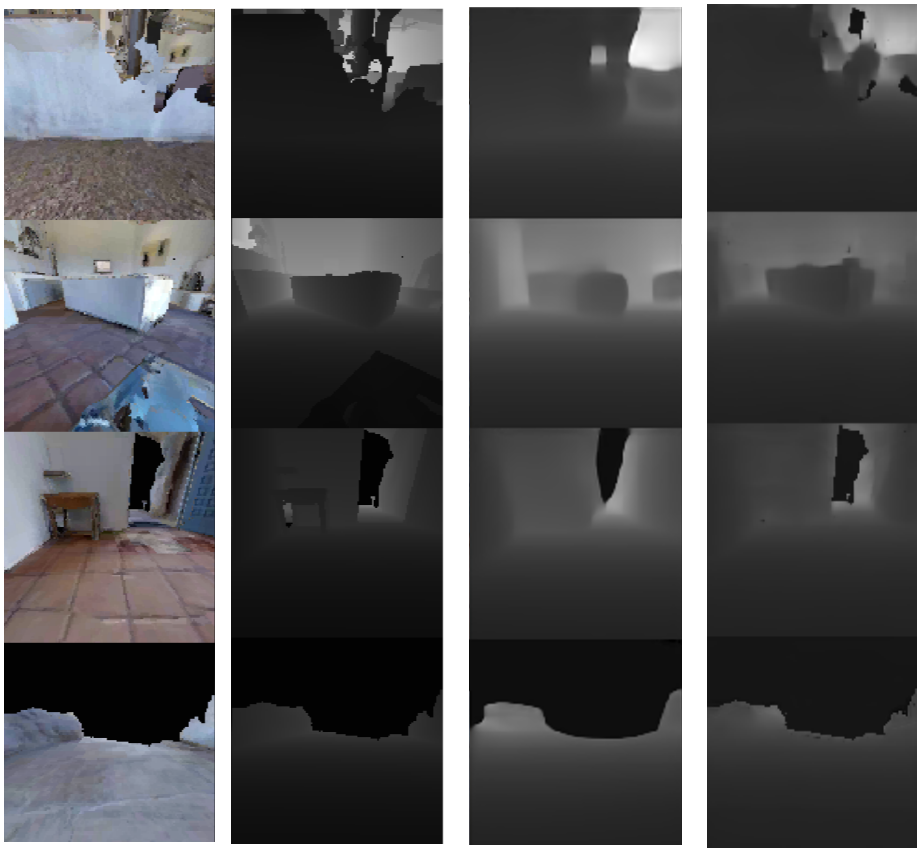


Figure 4.3: Input RGB image, ground truth depth, predictions without skip connections, and predictions with skip connections.

4.2.2 Transfer Training & Datasets

When transferring to the downstream depth estimation task, we freeze the encoder parameters but leave the decoder to be tuned. The learning rate is set to 0.001, the batch size to 16, and the training duration to 100 epochs. The training and evaluation datasets both comprise of samples from 15 unique environments (distinct from the RL training and evaluation environments). 500 viewpoints were sampled from each environment, for the total of 7500 samples in each dataset. We use L1 loss to compute the error in depth at the per-pixel level, where any invalid pixels with indeterminate depths are masked out. Exponentiated losses like L2 were trialled as well, but we find that the exponentiation overly magnifies the outlier predictions and negatively affects the results [4].

As mentioned previously, a larger baseline training dataset was generated for pre-training the agent's visual encoder in **Experiment B**. This was done with 500 samples from 75 environments, totalling 37,500 samples. These environments are again distinct from the RL environments and the transfer learning evaluation dataset.

4.2.3 Transfer Performance Measurement

For measuring the transfer performance to depth estimation, we compare the performance of individual RL agents to standardized baselines of frozen and unfrozen scratch encoders. The frozen encoder's random weights are preserved but the unfrozen encoder is allowed to tune its parameters, leading to better performance on the loss. We use Equation 4.1 to evaluate an agent against the frozen and unfrozen scratch models, where m stands for the metric (L1 loss on pixelwise depth estimation). This is called this an agent's *relative transfer performance* (RTP).

$$RTP(agent) = \frac{m(frozen) - m(agent)}{m(frozen) - m(unfrozen)} \tag{4.1}$$

Specifically, we take the average evaluation loss of the last 40 transfer training epochs, which is approximately when transfer training has converged in all runs. To mitigate effects from the stochastic nature of training, $m(frozen)$ and $m(unfrozen)$ are computed as the average of three runs for the frozen and unfrozen encoders. The individual agents' encoders were not able to be evaluated multiple times due to time constraint, however it is observed that the variations between repeated transfer trials are not significant. However, we do evaluate a single agent's encoder at various checkpoints of the RL training to record any variations as the agent approaches and proceeds past convergence.

[4] The loss comparison between L1, L2, and L4 [69] losses were performed by another student.

Chapter 5

Experiments

Now that the motivation and technical implementation have been established, we move onto the two sets of experiments that each investigates a respective direction of the APL. This chapter provides an overview of every embodied task trained and Chapter 6 cover the qualitative and quantitative results and observations.

5.1 Experiment A: Effect of Action on Perception

To reiterate the objective, we want to probe the embodied task parameters through varying the reward on specific axes and observe the downstream transfer performance of the visual systems that develop. These axes of variations can be either intra-task (changing the high-level task objective) or inter-task (changing implementation details of the task). As mentioned, Habitat-Sim does not support interactivity, so the domain of embodied tasks is restricted to either motor movement, visually-guided navigation, or localization tasks. Movement tasks do not have any navigation components, as the agent just needs to execute a series of motor actions. Visual search tasks, however, require some level of visual information processing, whether for identifying targets, landmarking, avoiding collisions, and/or path planning.

An embodied task is parameterized by a reward function, and as such each embodied task we implement falls into one of the following task objectives: *Random, ForwardMove, ForwardNav, ObjectNav, PointNav*, and *PathIntegration*. Figure 5.1 shows the progression of the task objectives from easy to difficult as well as the high-level skill dimension introduced for each. While we operate within a finite list of task objectives, a single objective, such as *ForwardMove*, can be defined through an infinite combination of rewards values. Using defined task objectives helps with organizing our experiments to align with existing benchmark experiments in embodied RL research.

For specific training parameters and inter-task reward variations of each embodied task trained, refer to Table 5.1. In total, we train 15 embodied tasks distributed across the six aforementioned task objectives – each one is described in the next paragaphs. The inter-task reward variations fall into four categories: collision-based, which penalizes for environment collisions; destination-based, which rewards

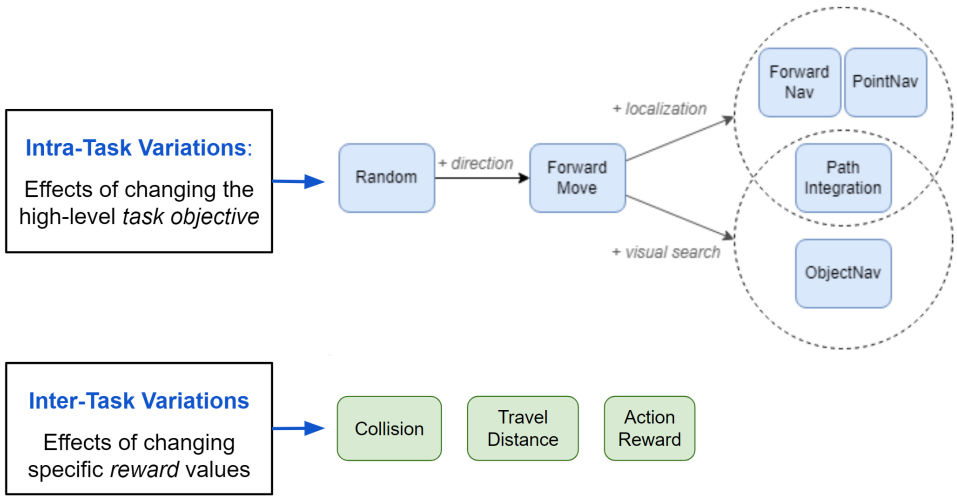


Figure 5.1: Intra- and inter-task variations.

for reaching a certain position or semantic goal and is usually the termination condition; action-based, which rewards taking specific actions; and object-based, which only exists for *PathIntegration* as a reward for finding specific objects in the environment. Not included in Table 5.1 is the living penalty, which is -0.01 deducted at each timestep for all agents. Some preliminary tests showed a living penalty may help with convergence, but its effect was not further investigated.

Random: We start by considering an entirely passive agent, one that receives random rewards at every step such that it cannot learn to increase its rewards. The agent keeps taking random actions and does not converge to any optimal policies (since there isn't one), but it still receives visual information. A variation is rewarding taking random actions, *Random Act*, where the agent receives a reward at each step if all previous actions in the episode are equally distributed. This is an optimizable goal, so we can compare if a convergence in the rewards effect the agent's learning abilities. Also categorized into this objective for convenience is the *Turn* agent, which is simply rewarded for choosing the *left turn* action. Altogether, this category of agents are for evaluating if random or turning movements can support depth perception development.

ForwardMove: An incremental skill is to train the agent to reach a forward destination. The basic *Forward 3m* task can be achieved without any visual input at all, as it only requires to select *go forward* until it reaches its 3m destination, upon where the episode would terminate. By moving in a straight line, the agent should receive strong motion parallax depth cues, thus this objective investigates if the mere presence of depth cues induces learning. Since the training is fast and straightforward, we also try several reward variations, such as adding an action-based reward (*Forward Act 3m*) increasing distance of the destination (*Forward 5m*), and using a collision penalty (*Forward 3m Collision*). The most significant is the collision penalty, as we reason that the agent needs to understand depth to some degree to prevent collisions (although it is also possible for the agent to learn an internal model of the environment and which areas to avoid).

ForwardNav: The next skill to integrate is navigation. Instead of moving forward in a clear path, the *ForwardNav 3m* agent now must navigate to a forward location 3m in front of its spawning location, but its path may be interrupted by furniture or

Task Objective	**Task Name**	**Rewards**			
		Action	*Collision*	*Destination*	*Object*
Random	Random	0	0	0	0
	Random Act	0.1	0	0	0
	Turn	0.1	0	0	0
ForwardMove	Forward 3m*	0	0	25 (3m)	0
	Forward Act 3m	0.1	0	25 (3m)	0
	Forward 3m Col.*	0	-0.5	25 (3m)	0
	Forward 5m	0	0	25 (5m)	0
ForwardNav	ForwardNav 3m	0	0	25 (3m)	0
	ForwardNav 3m Col.*	0	-0.5	25 (3m)	0
ObjectNav	ObjectNav	0	0	25 (door)	0
	ObjectNav Col.*	0	-0.5	25 (door)	0
PointNav	PointNav	0	0	25 (coord)	0
	PointNav Compass	0	0	25 (coord)	0
PathIntegration	PathIntegration*	0	0	25 (home)	5/obj
	PathIntegration Col.*	0	-0.5	25 (home)	5/obj

Table 5.1: Reward parameters of all embodied tasks trained in **Experiment A**, where * denotes the training was repeated in **Experiment B**

walls. This type of navigation is based on localization, not visual search. It is more difficult to solve and absolutely requires visual input, as changes in the RGB input is the only way the agent can know if its path is blocked or not (the agent doesn't get its location). As with the *ForwardMove* objective, we test out a variation with an added collision penalty *ForwardNav 3m Collision*.

ObjectNav: *ObjectNav* (object-goal navigation) is a task introduced by [70], which specifies the agent's goal as navigating towards an instance of a given object class. The task is notoriously difficult to optimize, so we do not focus on solving it at a high level of complexity. In traditional implementations, the agent may be asked to find an object from any class from a predefined list. We change it to an easier termination condition, where reaching any 'door' would be counted as success. 'Door' is chosen as the target semantic class since it is the only one that is present in all training and evaluation environments. Again, a collision variation *ObjectNav Collision* is trained.

PointNav: *PointNav* [1] (point-goal navigation) is defined by [70] as navigating towards a given coordinate location. We attempt two variations of the task: a) using a unique coordinate but giving the agent access to a 'compass' direction pointing towards the destination alongside the state information (*PointNav Compass*), and b) giving the agent a unique coordinate as the only input (*PointNav*). Since these are technical design choices in the architecture and input, we do not consider these as reward variations. The baseline version of *PointNav*, where the same coordinate is given each time, can be thought of as *ForwardNav*.

PathIntegration: Lastly, we introduce a biologically-inspired task called *PathIntegration*, which is the mechanism used by animals to return home [71]. It is shown that animals that take complex paths in their journey can still return home using a direct route, indicating that they keep track of their position and orientation even without visual cues [72]. To add in a visual search component, we place a number of 'food' objects in the environment within a certain radius of the starting position. A few variations were experimented with, but it was decided that 2-7 meters search radius presented a suitable challenge to the agents. The agent receives rewards for collecting one or more 'food' objects, and upon doing so, are rewarded additionally if they return home. We also test with a collision penalty *PathIntegration Collision*.

5.2 Experiment B: Effect of Perception on Action

The second stage of experiments focus on completing the APL through testing if the pre-encoded knowledge of depth would benefit training active agents. The setup is simple, as described in Chapter 4, the visual encoder in the agent is pretrained on depth estimation such that the agent commences training with a good understanding of pixel-wise depth.

Due to time constraints, not every embodied task was repeated with the pretrained depth visual encoder. Complex training regiments, such as for navigation tasks, takes several days to a week to reach convergence, and the policy learnt may not even be completely optimal. We therefore selected only six agents to repeat with depth pretraining: *Forward 3m, Forward 3m Collision, ForwardNav 3m Collision, ObjectNav Collision, PathIntegration*, and *PathIntegration Collision* agents to repeat with the depth pretraining (these agents are indicated with * in their names in

[1] PointNav is implemented and trained by another student.

Table 5.1). These represent a variety of tasks objectives that require either no visual information, or some form of visual information for collision avoidance, navigation, and object search.

The visual encoder was allowed to be tuned during the agent's training. This was done to observe how the agent's perceptual requirements might deviate from its initialized state of optimized pixelwise depth prediction. Of course, our analysis cannot fully reveal what perceptual tasks the trained encoder has reweighted itself to optimize performance for, since we lack the ability to evaluate on other tasks and metrics. However, if the downstream transfer performance deteriotes, this would indicate that good depth estimation is not necessary or optimal to perform our set of embodied tasks.

According to [41], metrics for comparing improvements in RL performance include reward jumpstart, asymptotic performance, and time to threshold. These are illustrated graphically in Figure 5.2. In particular, jumpstart and time to threshold would indicate improvements in sample efficiency. Higher asymptotic in the evaluation environments would indicate better generalization to unseen environments.

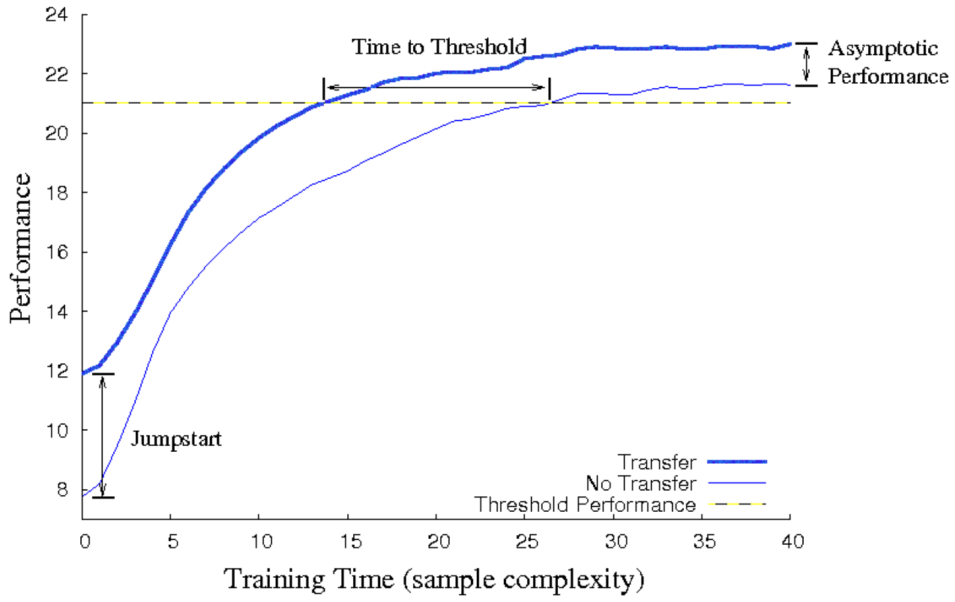


Figure 5.2: Measurable improvements in RL training from [41]

Chapter 6

Results

6.1 Experiment A: Effect of Action on Perception

This section showcases the primary results of **Experiment A**. The relative transfer performance (RTP) of each agent is recorded after its training converged [1]. This is can be seen in Figure 6.1, which organizes the task objectives with colour coding.

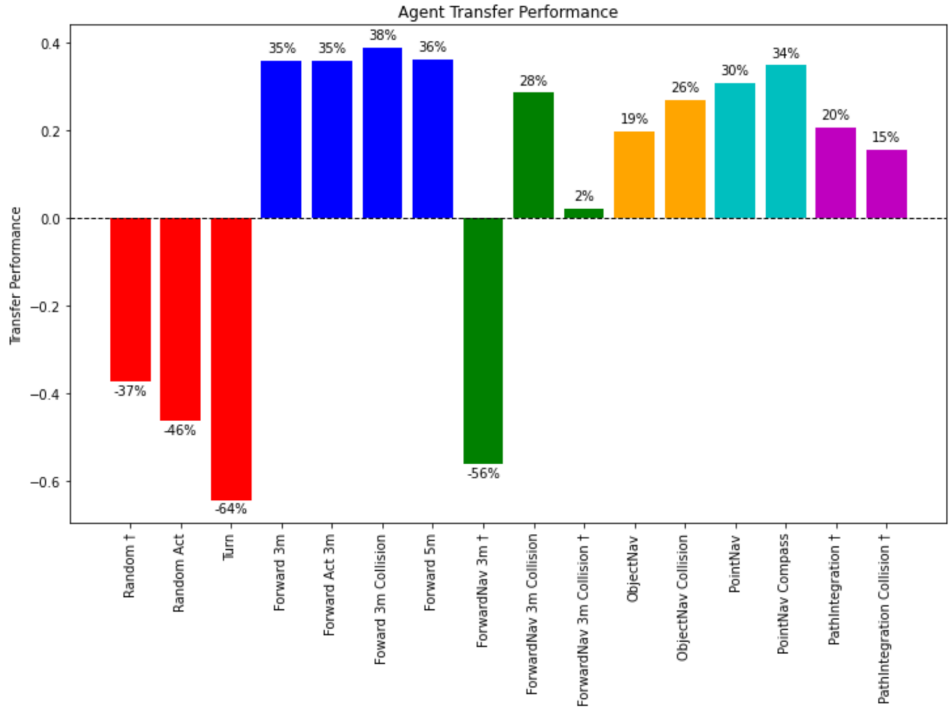


Figure 6.1: Relative transfer performance of each agent in **Experiment A**, where †denotes agents that did not reach convergence or finish training.

In summary, the range of RTPs spans from -64% worse than scratch (*Turn*) to

[1] In most cases, we attempt to train the agent to convergences, other in situations where the reward cannot be optimized or the task is too complex to fully achieve within a few days.

38% better than scratch (*Forward 3m Collision*). Figure 6.2 shows the outputs generated by the best and worst models on evaluation images. The best model is able to capture detailed depth discrepancies of the room better, such as the kitchen island in the 2nd row and the extended back hallway in the 3rd row.

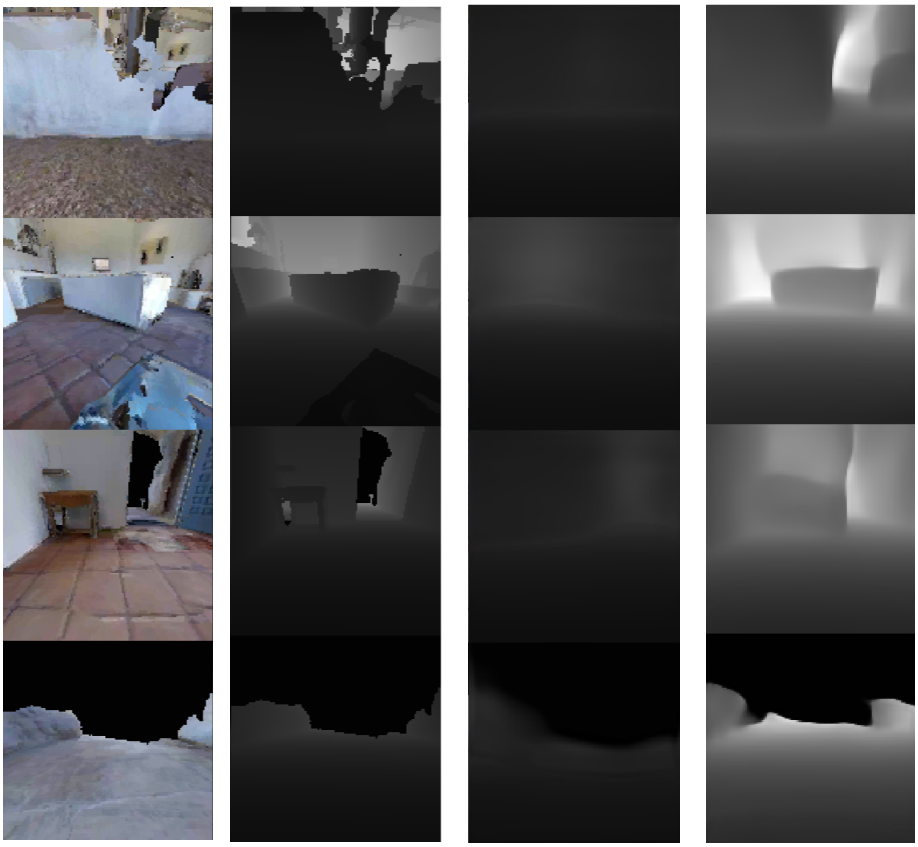


Figure 6.2: Input RGB image, ground truth depth, *Turn* agent predictions (worst), and *Forward 3m Collision* agent predictions (best).

It should also be noted that due to the inherent randomness and instability of RL, our reported numbers would likely fluctuate using other seeds. As an example, *ForwardNav 3m Collision* was performed twice. The first runthrough yielded a good policy with high RTP, but the second runthrough did not converge well and resulted in a negative RTP. It is not clear how robust these data points are as training each agent takes far too long to repeat multiple times. However, given that the nature of some of the tasks are very similar (such as *Forward 3m* and *Forward 5m*, which resulted in similar RTPs), they can serve to provide an approximate estimate of intra-task variance.

We strive to evaluate each agent approximately at the checkpoint that its rewards in the evaluation environment reached convergence, but this method was not rigorously enforced since it relies on estimating visually if/when the agent has converged. Some agents (*PointNav, PointNav Compass*, and *ForwardNav 3m Collision*) were trained under the 15/0 environment splits, which is noteworthy as a higher number of training environments speeds up training and improves overall performance. The *PathIntegration* agents were kept to a 10/5 environment split, but it became evident that training would take too long, so their training runs had to be terminated early.

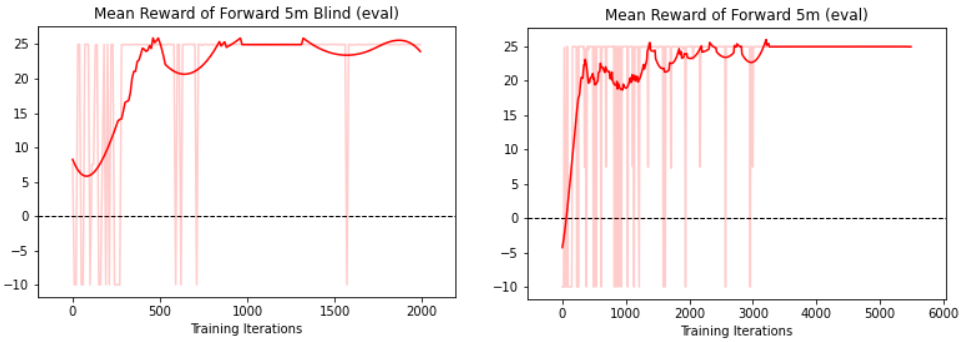
6.1.1 Effect of Task Objectives

This section briefly discusses and patterns observed at the task objective level.

Random: Categorically, the *Random* agents performed the worst in terms of RTP, with all three agents being considerably worse than frozen scratch. The optimizable version, *Random Act*, did not contribute any improvements. It is interesting to note that the *Turn* received the worst RTP score at -64%. It was initially suspected that the turning angle of 30°is too large for dynamic depth cues to be coherently preserved from frame to frame. If the jump in angle is too big, then the agent won't be able to learn from the motion parallax. However, this is actually disproven by re-training the *Turn* agent with a 5°turning angle, but no improvements in RTP were observed.

ForwardMove: Somewhat unexpectedly, the simple *Forward* agents categorically outperformed all other agents with RTPs between 35 - 38%. With the exception of *Forward 3m Collision*, none of the other agents actually require any depth understanding to develop optimal policies. A theory to present here is that forward movement may be much richer in depth cues than turning movements, which is why *Forward* agents are better at depth estimation than *Turn* agents. This can be explained by that depth cues from motion parallax are very strong in an agent that executes linear motion in empty spaces, which puts the agent within view of room furnishings, doorways, and floor edges, which are all feature-rich.

To further explore the role of vision on training forward movement, we compare the training a blind and a sighted agent on *Forward 5m*. Their RL performance curves are shown in Figure 6.3, demonstrating that the blind agent was able to converge within a shorter training period. Therefore visual information could possibly be a deterrence for learning the task rather than an asset, although this could also be attributed to initialization.



(a) Blind agent. (b) Sighted agent.

Figure 6.3: Comparison of RL training performance of a blind and sighted agent on *Forward 5m*, where the bottom axis is training iterations.

Taking it one step further, we resumed the converged blind agen as a sighted agent (as in the training was resumed post-convergence with RGB inputs). The progression in the RTP at various checkpoints of the blind training and the resumed training are shown in Figure 6.4. The policy of the agent did not observably change, yet the RTP immediately improves after visual input is reinstated. This means that the visual encoder is undergoing changes while the policy is not, perhaps pointing to distinct visual and motor update mechanisms. This also indicates that the mere

presence of good depth cues can induce learning, even after convergence when there is no longer need to learn anything. The mechanism which determines how the agent tunes its visual encoder is not fully understood in the context of this project, but can potentially be explored further.

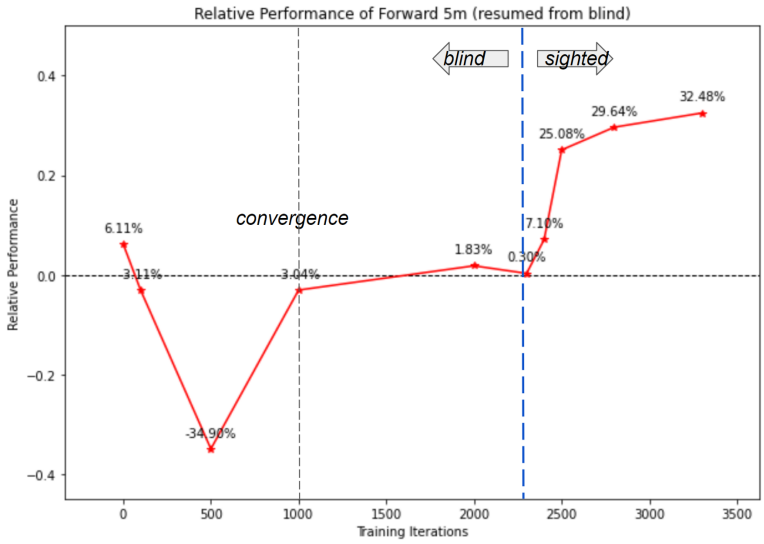


Figure 6.4: Progression of the RTP of an agent resumed from blind.

ForwardNav: *ForwardNav* was a challenging task to train and resulted in highly varied RTP results from -56% with *ForwardNav 3m* † to 28% with *ForwardNav Collision*. The agent is spawned at random locations in the environment without verifying if it has a clear path towards its destination and if it needed to navigate around blockages and walls. Generally, the agent is able to solve simple variations of this task, but not when it has to traverse through multiple rooms or make large detours. As state previously, we tried training multiple times, but only converging to a good policy in one instance of *ForwardNav Collision*. The non-convergent agents had sub-optimal policies that resembled *Random* agents, which means they did not receive good depth cues from their erratic motion. No conclusions can be confidently drawn from *ForwardNav* other than the quality of depth cues seems to be the strongest determinator of RTP.

ObjectNav: As a reminder, *ObjectNav* agents were trained to find doors, upon doing which yields a successful reward and termination of the episode. While both agents yield satisfactory performances, their 19% and 26% RTPs are better than scratch but worse than most other agents. A problem identified in the problem set up is that the number of doors in the environment is highly excessive. Figure 6.5 depicts three sample bird-eye-view maps of Gibson training environments, where each coloured marker represents a door. As seen, an agent spawned in the environment would be likely to be nearby a door, resulting in a short and simple trajectories. It is not clear if the trained agents actually learned to find doors, or simply to do enough exploration to serendipitously stumble across doors.

Comparing the training and evaluation environment performances of *ObjectNav Collision* in Figure 6.6, it is evident that the agent is overfitting to the training environments and generalizing poorly to evaluation environments. A combination of the short travel trajectories and the generalization problem can reduce the quality of the depth cues and affect perceptual learning. In general, none of the other navigational agents (*PointNav* and *PathIntegration*) outperform *ForwardMove* agents

Figure 6.5: Bird-eye-view of Gibson environments and location of doors.

in terms of RTP at the early stages of convergence.

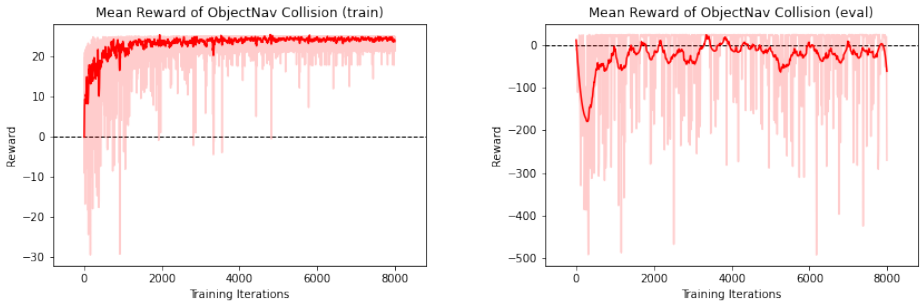


Figure 6.6: Comparison of RL training and evaluation performances of *ObjectNav Collision*.

PointNav: *PointNav* theoretically should have been similar to *ForwardNav*, but the two trained agents actually reach high RTPs of 30% and 34% with the added compass. A contributing factor here might be that 15 training environments were used, so it's difficult to project how much RTP would degrade if training was repeated with 10 environments. *PointNav* was also the only task objective that received any additional inputs – the destination location and the optional compass travel direction. Again, it's hard to quantify how much this change might have impacted the RTP. The *PointNav* agents are explored more in Section 6.1.3, which looks at RTP trends post-convergence.

PathIntegration: As stated previously, a huge limiting factor in evaluating the *PathIntegration* agents is that neither had time to fully converge. Rather, they were trained to 7000 iterations, a point where noticeable improvements in rewards stalled. This could mean that the task is too difficult to fully optimize, but it's also possible that sudden policy improvements can happen later on. To compare, the *PointNav* agent trained on 15 environments took 8000 iterations to converge.

Through observing recordings of training episodes, we see that the agents are able to find food objects successfully, but struggle more with returning home. In some cases, they are able to return by seemingly random exploration. Their movements are still erratic at times, which we now know reduces the quality of depth cues. Thus, it then makes sense that the RTP of pre-convergence *PathIntegration* agents are better than scratch but still at middling levels.

6.1.2 Effect of Inter-Task Reward Variations

Action Rewards: The embodied tasks denoted with "Act" (*Random Act* and *Forward Act 3m*) gave an additional reward for selecting the correct action at each step. This test is to probe if rewarding motor actions would lessen the involvement of the visual system. The *Random Act* agent gives inconclusive results as it likely did not see meaningful depth cues, but the *Forward Act 3m* agent is slight worse than *Forward 3m* despite converging faster. This slight discrepancy may be showing the shift of attention from visual to motor information, but is not significant enough to draw conclusive results.

Collision Penalty: The collision axis of variation compares if sensitivity to environmental collisions would heighten the agent's awareness of depth. The results are somewhat conflicting as collision penalties improved the RTP of *ObjectNav*, *Forward 3m*, and *ForwardNav 3m*, but lowered the RTP of *PathIntegration*. However, these differences are either very minute or can be affected by initialization. In addition, *ForwardNav 3m* was highly difficult to train and we are not able to determine the conditions for why some runs succeeded and others failed. While depth perception is logically seen to be useful for collision-avoidance and some initial results seem to support this theory, more repetition is needed.

Travel Distance: The only explicit distance comparison experiment we executed is *Forward 3m* and *Forward 5m*, which yielded very little difference in RTP – 35% and 36% respectively. However, there is some indication that longer travel distances, especially in the context of navigation, requires more intelligent policies. We discussed that *ObjectNav* has shorter trajectories and lower RTP in comparison to other tasks. In a prior experiment, *PathIntegration* was tested with a smaller search radius of 0-5m, which means the agent does not have to go very far or remember its location, as random exploration can result in high returns. As follows, analysis of the agent's visual saliency and RTP to depth estimation showed poor results [2].

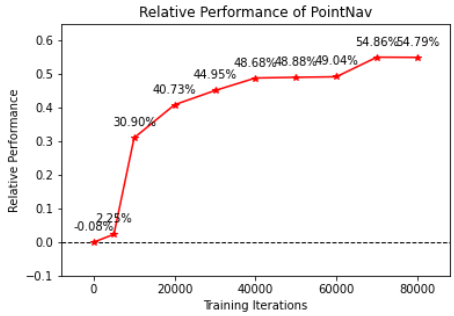
6.1.3 Perceptual Learning Post-Convergence

In addition to evaluating the RTP of each agent at a singular post-convergence checkpoint, we actually evaluated each agent at multiple checkpoints pre- and post-convergence to observe how depth perception abilities develop alongside RL training. Figure 6.7 shows an example of progressive RTP evaluations of the *PointNav* and *Forward 3m* agents from the 0th iteration to some number of iterations past convergence [3]. When contrasted against the RL performance curves, it is evident that RTP can keep improving even after RL rewards plateau.

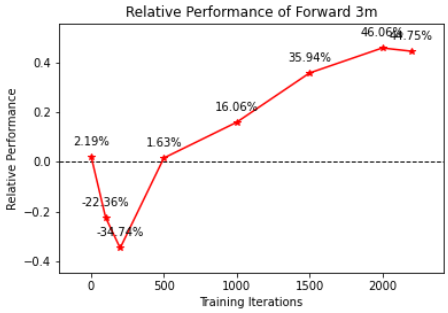
More concretely, Figure 6.8 shows the post-convergence changes in RTP of all agents that exhibited positive RTPs (so excluding the *Random* agents). Note that *PathIntegration* agents are also included here despite not fully converging, but their RL performance curves did show plateauing. Quite remarkably, 9 out of 11 agents showed positive improvements in post-convergence RTP. The most significant change is the *PointNav* agent, which improved its RTP relatively by 23% and was trained up to

[2] Quantitative results not included as this agent was trained long before the semester project began.

[3] "Convergence" is defined flexibly in this context, it just means approximately when the rewards begin to plateau.

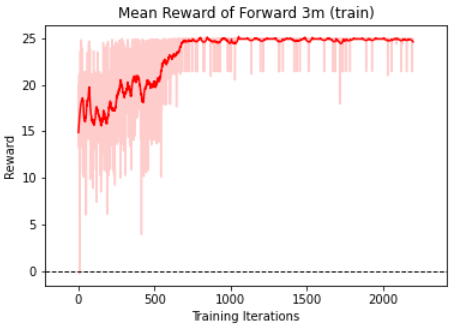


(a) *PointNav* RTP.



(b) *Forward 3m* RTP.

(c) *PointNav* RL performance.



(d) *Forward 3m* RL performance.

Figure 6.7: Progression of RTPs and RL performance for *PointNav* and *Forward 3m* agents.

80,000 iterations, significantly more than any other agent. This is certainly an interesting observation, but the potential of other agents' RTPs to improve cannot be estimated since the mechanism of perceptual development is not well understood. Will they reach the same RTP as *PointNav*? Or is the best possible RTP attained with an optimally-trained agent constrained purely by the task requirements? E.g. an agent that makes use of depth information that has been fully optimized would have better RTP on depth estimation than an agent that doesn't require depth.

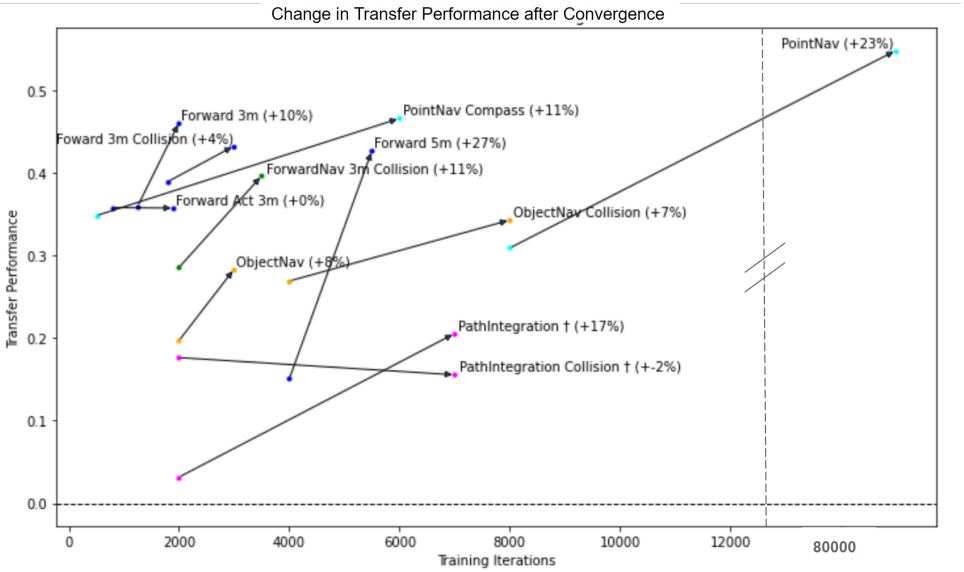


Figure 6.8: Changes in the RTP of all agents post-convergence.

One contributing factor to consider is the smoothness of the trajectories, as we know that forward movements generate better depth cues for learning than erratic motion. This skill would not be reflected well through the raw reward values, as the only reward that comes into play is the living penalty, which is very small. A long trajectory filled with unnecessary motions would only receive a slightly lower score than a fast, direct trajectory. A better way to distinguish this in the future is to compute the path length of the agent, such as using the 'Success weighted by Path Length' (SPL) metric showing in Equation 6.1. Under our logging capacity, the path length was not implemented initially, so it would not make sense to redo a large number of trials to capture this statistic. The information given by SPL can be useful for finer grained analyses in the future. Thus it is conceivable that the *PointNav* agent gradually improved the efficiency of its policy while its rewards seemingly plateaued.

$$SPL = \frac{1}{N}\sum_{i=1}^{N} Success_i \frac{shortestPath_i}{max(shortestPath_i, agentPath_i)} \tag{6.1}$$

6.2 Experiment B: Effect of Perception on Action

This section examines the results of **Experiment B**, the RL training performance of agents with and without the pretrained visual encoder. As mentioned in Chapter 5, the metrics we are interested in are jumpstart, asymptotic performance, and

time to threshold. The experiment was performed on six tasks: *Forward 3m, Forward 3m Collision, ForwardNav Collision, ObjectNav Collision, PathIntegration*, and *PathIntegration Collision*. Figure 6.9 shows the comparison of RL rewards by depth-pretrained (in blue) and scratch (in red) active agents. All graphs are from evaluation environments except *ForwardNav 3m*, which was trained with a 15/0 split. The last row also shows the training environment results of *PathIntegration*, which is noted to exhibit patterns of overfitting.

Visual inspection of the reward performance curves is used as the primary evaluation method. While computational methods can be used as well, a limitation is how well the smoothing function (a handtuned Savitzky–Golay filter) represents the original data. Most of the curves are extremely noisy, making it hard to standardize evaluation criteria (e.g. which region counts as asymptotic performance?)

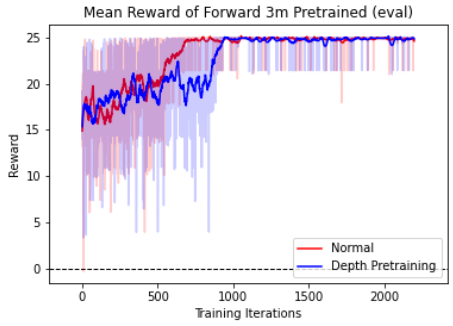
Neither of the *ForwardMove* agents saw improvements, in fact, pretraining produced slower time to threshold and initial lag in performance in both. The asymptotic results between the pretrained and scratch agents are comparable. *ForwardNav Collision* and *ObjectNav Collision* showed clear improvements in the jumpstart performance in the initial training region, where most of the negative rewards can be attributed to collision penalties. Depth pretraining thus can be applied to improve the agent's ability to learn collision avoidance early on. *PathIntegration* and *PathIntegration Collision* showed comparable performance in the evaluation environments between the pretrained and scratch agent, but the jumpstart improvement is more pronounced in the training environment. The overally asymptotic performance is much better in training than evaluation as well, indicating these two agents experienced significant overfitting, and depth pretraining may be exacerbating it. The only task that resulted in an improvement in asymptotic performance and time to threshold is *ObjectNav Collision*.

6.2.1 RTP Progression

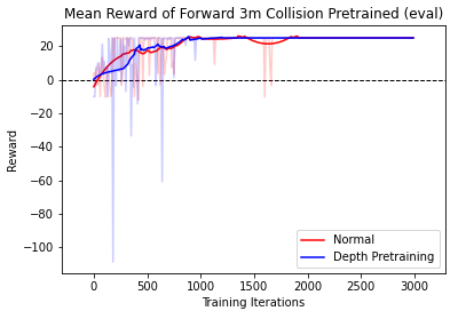
Again, we looked at the progression of RTPs of the depth-pretrained agent as it learns the embodied task. How the visual encoder would continue to perform on the depth estimation task is not intuitive to predict. Since the encoder was allowed to update its weights, we believed it was unlikely that pixelwise depth estimation would be fully preserved. Figure 6.10 show the depth pretrained agent's RTP in blue and the scratch agent in red for the same six training tasks.

The global pattern which emerges immediately is that the pretrained encoder regresses while the scratch encoder improves, often converging towards the same or very similar depth estimation abilities. This seems to suggested that the training process pushes the visual encoders towards a set of perceptual abilities that include depth understanding, but not precise depth estimation. Intuitively this makes sense as biological agents do not need the ability to estimate distances accurately. *Objectnav*'s pretrained encoder is the only one to maintain an improvement in RTP over scratch. This can suggest the optimal level of depth estimation performance is higher than what the scratch agent was able to learn.

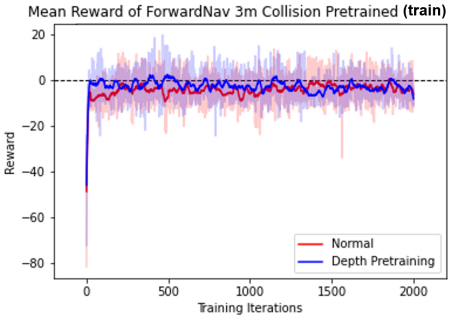
An alternative theory may be that the pretraining loses its effect very quickly, and the encoder reweighting is dominated by learning to process the input images. This can explain why the RTP of the pretrained encoder undergoes a rapid drop within the first 500 iterations of training. Since RL training is highly unstable at the beginning, the visual encoder might also be undergoing significant changes to



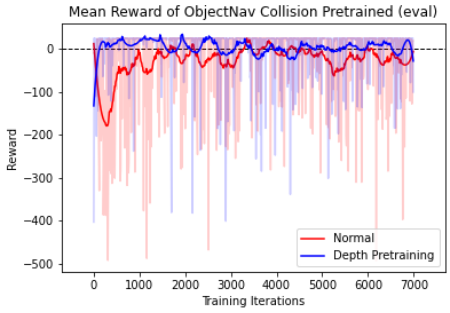
(a) *Forward 3m* (eval).



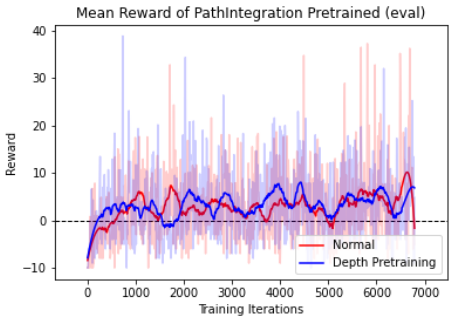
(b) *Forward 3m Collision* (eval).



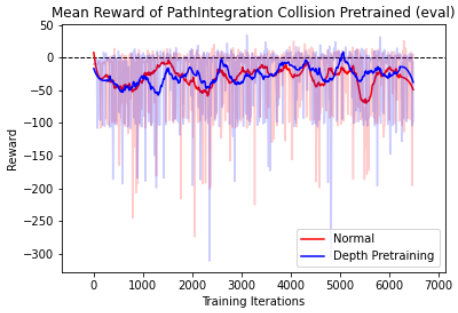
(c) *ForwardNav 3m Collision* (train).



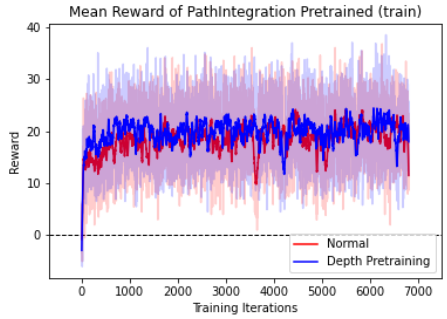
(d) *ObjectNav Collision* (eval).



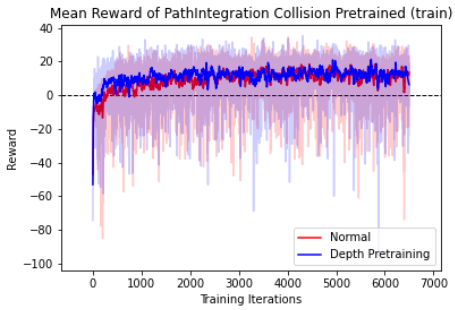
(e) *PathIntegration* (eval).



(f) *PathIntegration Collision* (eval).



(g) *PathIntegration* (train).



(h) *PathIntegration Collision* (train).

Figure 6.9: RL performance curves of depth-pretrained (blue) and scratch (red) agents.

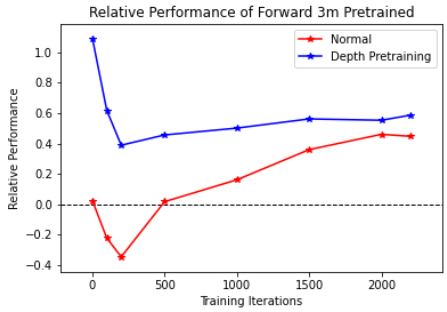
become similar to the scratch encoder.

6.3 Alternative Pretraining Methods

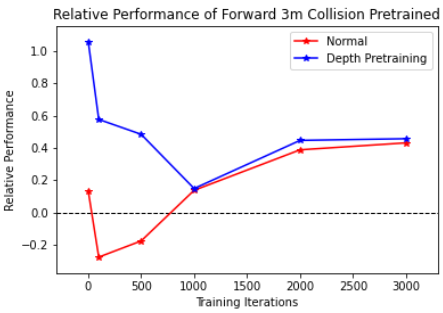
This is a side experiment that spawned out of the limitation of using depth estimation as the pretraining task and the uncertainty about how the visual encoder learns during training. The question here is what pretrained encoder can be used in lieu of the depth encoder? While we don't have any perfect visual systems to sample from, there is the encoder from the *PointNav* agent that has been trained to over 80,000 iterations. It has the best RTP and its policy is observably smooth, so it serves as a good substitution for the optimal visual system for navigation.

We retrain the *PathIntegration Collision* agent with the *PointNav* encoder. Despite the task objectives being different, both involves localization. The evaluation and training performance curves for the agent trained with the *PointNav* encoder is shown in the left column of Figure 6.11. Similar to what we saw with the depth-pretrained agent, the evaluation results are noisy and show ambiguous results, but the training environment see a clear improvement in jumpstart and asymptotic rewards. This would indicate overfitting to the training environments. Since the *PointNav* agent was trained without evaluation environments, we can't determine the degree of overfitting of the original encoder, which is problematic and voids these results.

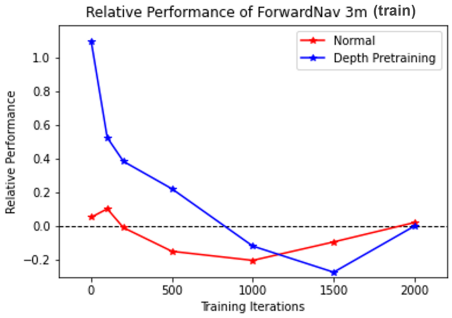
Another study [48] freezes the visual encoder after transferring it to the RL agent, showing improvements in training speed and generalization. We also attempt with frozen weights for the depth encoder and the evaluation and training results are show in the right column of Figure 6.11. While the evaluation performance again is similar between the pretrained and scratch agents, the training performance of the frozen pretrained encoder actually underperforms scratch, but matches its respective evaluation performance. This indicates the frozen pretrained encoder is less prone to overfitting, but performance improvements over scratch is still not observed. More experimentation would need to be done with freezing the encoder to understand its effects.



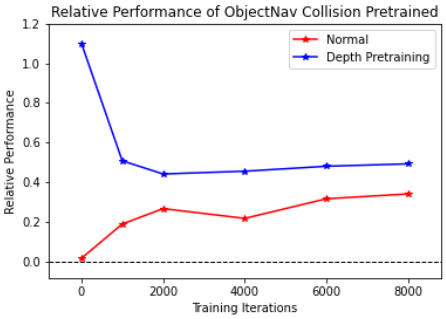
(a) *Forward 3m.*



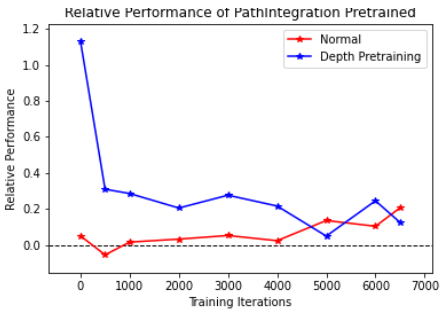
(b) *Forward 3m Collision.*



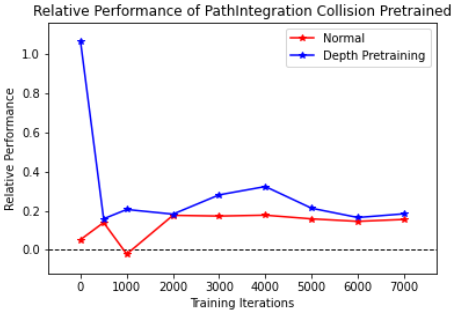
(c) *ForwardNav 3m Collision.*



(d) *ObjectNav Collision.*

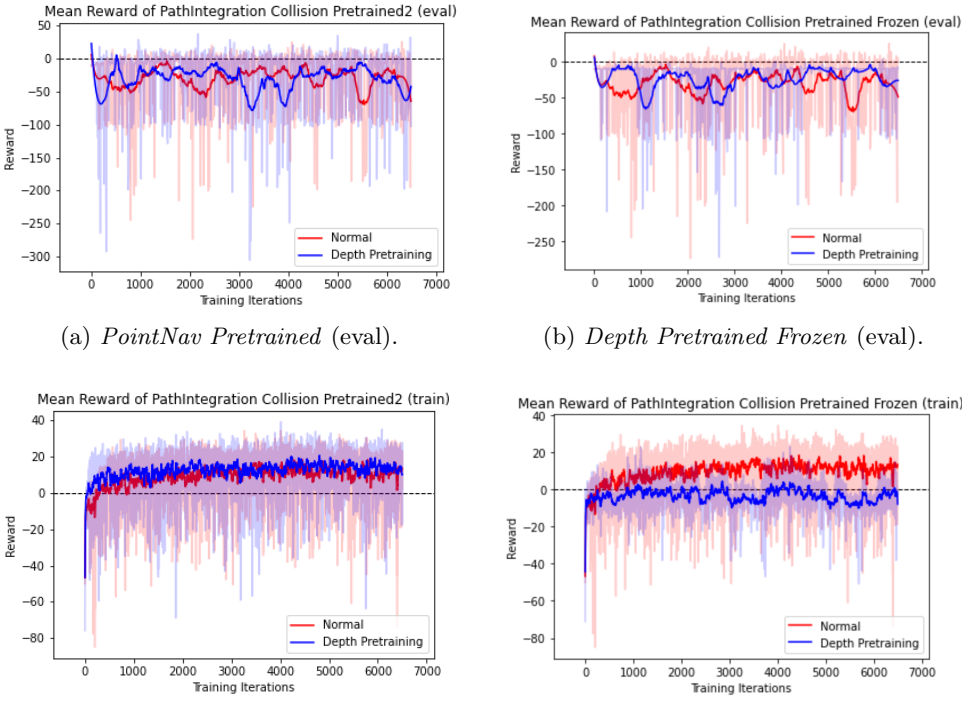


(e) *PathIntegration.*



(f) *PathIntegration Collision.*

Figure 6.10: Comparison of RTP progression for depth-pretrained (blue) and scratch (red) agents.



(a) *PointNav Pretrained* (eval).

(b) *Depth Pretrained Frozen* (eval).

(c) *PointNav Pretrained* (train).

(d) *Depth Pretrained Frozen* (train).

Figure 6.11: RL evaluation and training performance curves for *PathIntegration Collision* agent with a *PointNav* encoder (left) and frozen depth encoder (right).

Chapter 7

Discussion

The study presents an experimental forage into evaluating the perceptual abilities, specifically depth perception, of reinforcement learning agents trained on embodied tasks. This is an open-ended exploratory topic, as even perceptual learning and perceptual abilities are still continually investigated in humans. We take inspiration from the action-perception loop to propose this question, but it is clear that the experimental results are affected by computation limitations of RL training. Many factors combine to contribute to the development of depth encodings in RL agents, making them difficult to disentangle. To bring in practical applications of APL, we also demonstrate how pretraining active agents with perceptual priors can tangibly effect RL training. While depth estimation was not necessarily demonstrated to be the best skill to supplement our specific set of embodied tasks, the approach can be promising for tackling pervasive technical challenges in RL given an improved pretraining task and metric.

In **Experiment A**, we experimentally show that active agents encode depth information significantly better than scratch encoders, but the quality of depth cues affect the depth transfer abilities. Random and turning policies notably underperform scratch, indicating that turning movements are not optimal for generating learnable depth cues. Within our limited RL training framework, the agents that exhibited the best depth transfer performance are the simplest *ForwardMove* agents, which violated initial expectations as complex navigation agents were presumed to learn more depth to fulfill their task requirements. However, our experiments show that through simply receiving passive visual inputs while moving through an environment, agents are able to encode perceptual information that are not required for their policy (recall the blind agent that was resumed as a sighted agent in Figure 6.4).

The contributions of other task variations are not easily isolable due to many agents not converging to an optimal policy and therefore not being directly comparable to each other. With the potential exception of collision-avoidance, none of the other inter-task or intra-task skill variations are conclusively shown to produce better depth perception than *ForwardMove*. To verify any significant contributions of collision (or other task variations) on perceptual development, more trials of both RL training and transfer learning need to be performed to quantify their respective variability. More thought should be put into differentiating the parameters of intra-task variations and being clear on what embodied skills each task objective trains. For example, *ForwardNav* and *PointNav Compass* are very similar behaviourally,

so they should theoretically produce similar visual systems. Any discrepancies may be artifacts from the quality of training and the reward function that the task is parameterized by.

Observing the post-convergence perceptual learning is also an unexpected finding – either indicating our measure of convergence is inadequate or that the mechanisms of perceptual learning and policy learning are more distinct than originally thought. An cognitive science perspective about perceptual learning is explored in Section 7.2. To better evaluate the quality of an agent's policy, we can adopt a more informative RL training metrics like SPL (Equation 6.1), which accounts for the length of the path taken in addition to the success of an episode. We can also reweigh the reward values to penalize for longer paths by using larger living penalties.

Experiment B showed that agents trained on more navigationally complex tasks generally saw benefits in their jumpstart performance from depth pre-training in comparison to simpler *ForwardMove* agents, but again we still lack significance in the results as several agents were limited by overfitting. The problem of overfitting can be mitgated by either increasing the number of environments that can be ran parallel or applying augmentation on the images. It is not shown that asymptotic performance or time to threshold is improved in the agents, with the exception of *ObjectNav Collision*. The visual encoders of both pretrained and scratch agents are shown to converge towards the same RTP, although it is unknown if this is due to a condition of visual optimality, or just the natural mechanisms of RL training that reconfigured the pretrained encoder to be like the scratch one.

[48] and [49] perform larger analyses on how encoding visual priors into the visual processing of RL agents impact their performance. The main difference is that the feature encoder for a given mid-level visual task is frozen during the agent's training. While we did test with a frozen depth-pretrained encoder, benefits in performance were not observed, although overfitting was reduced. [48] determines that the top visual features for navigation (similar to *ObjectNav*) are *object classification, semantic segmentation*, and *curvature*; and the top features for local planning (similar to *PointNav*) are *3D keypoints, normals,* and *curavture* again. The skill of *depth estimation* is actually best for *Exploration* agents, which is an embodied task that we did not test. Some of the top tasks are intuitive for the embodied task and other less so, which certify that we would benefit from expanding beyond depth estimating to include other visual skills.

7.1 Inadequacies of Depth Estimation

The potential irrelevance of depth estimation has been mentioned many times throughout this report, but a full discussion is warranted given that these experiments depended on depth estimation being a good task, which was not verified by our results. This led to ambiguity in the analyses as it became difficult to attribute how much of the depth perception abilities was learned because it was necessary to the policy, and how much was passively encoded simply as a result of the information being available – and what is driving each learning mechanism?

Depth was initially chosen for its easily accessible ground truth and its apparent role in ecological navigation. However, visual intelligence is much more complicated and may not be able to be broken down into perfectly orthogonal visual tasks bases which we can interpret. For example, the Taskonomy dataset [40] contains 26 mid-

level visual tasks, but many surely share the same underlying skills. There exist a tradeoff here as to if we should pursue breaking down visual skills into independent dimensions, or if we should build up a metric from a pre-defined set of known skills. The validation of either metric combination would be difficult as well, as logically a good visual metric should result in good performance from humans in addition to trained agents.

If there existed a more relevant perceptual task to evaluate on, we may see the discrepancies in the RTPs of agents trained on different tasks amplified, such as elevated performance of the navigation agents in comparison to the motor agents. [11] also evaluated on normals, classification, affordances, etc. Their experiments are not directly comparable since they train two agents adversarially in an interactive environment, which enabled agents to learn richer representations of their observations. The game of *Cache* required agents to play hide and seek with an object against each other. This added level of task complexity purportedly enabled agents to outperform navigation agents significantly on most downstream transfers, with varying levels of improvement between tasks.

7.2 Task-Irrelevant Perceptual Learning

This section briefly highlights an comparable learning phenomenon as what we observed with motor agents encoding passive visual cues despite not needing them. There is a term coined task-irrelevant perceptual learning (TIPL), which describes the ability to learn secondary task cues while training on a separate primary task [73]. It is proposed that the primary task cues and secondary task cues require different learning mechanisms which undergo differently staged processing in the brain, and it is unlikely that focused/sustained attention is needed [74]. However, this only occurs under the condition that the secondary cues are not strong enough to distract from learning the primary cues, which it would otherwise be suppressed by the brain for [75]. There is shown to be a positive correlation with learning the secondary tasks and the presence of rewards, even if said rewards are unrelated to the task [76].

How this plays into computational agents is not entirely clear, as it is not so simple to extend models of cognitive behaviour to artificial intelligence. There are certainly parallels to point out – namely, our agents learned depth (a secondary task with apparent visual signals) while training for a different primary task (embodied navigation) under the presence of a learning reward. However, human perceptual learning tests were never performed in the context of visuomotor learning, as the standard TIPL experiements isolate basic visual task dimensions. Something like depth and navigation is too intertwined into our daily arsenal of skills to properly investigate in humans through additional training. With that being said, chasing biologically plausible explanations for computational models and experiments may be a fruitless pursuit. Even if TIPL is observable in RL agents, the benefits of it are not clear pertaining to practical applications to RL. This leads to the next section, which discusses potential pathways of using the technical platform as an investigative tool for cognitive science instead.

7.3 RL Agents as Testbeds

Reinforcement learning agents, although limited in their generalizability and expressiveness in comparison to human cognition, can be used as a computational model in the field of psychology [77]. Like biological agents, they receive sensory inputs and use the information to choose an output action for accomplishing a goal [4]. Such agents can be seen as proxies to biological organisms that develop perceptual capacities through training on active tasks like embodied navigation, which is analogous in the biological domain to the task of *finding food or shelter*.

In alignment with previous works exploring the usage of artificial agents in the context of human cognition concepts [78, 11], we take this as an opportunity to add onto the body of literature. RL agents can be used to explore complex phenomenons in cognitive science and behavioural psychology that are difficult to test in humans. Neural networks (and other computational models) can be taken apart, probed, and run repeatedly with control.

We attempted to break down a high level embodied task into additions of basic task variations to isolate their individual contributions to the development of depth perception. While results lack significance and task variations cannot be fully disentangled to be orthogonal to one another another, an emerging picture is still painted that passive visual signals are encoded through movement and that depth is potentially useful for solving some embodied tasks. Under more life-like simulation conditions (interactive environment, binocular vision, richer action space, etc), we may be able to draw closer approximations to perceptual learning through behaviours of biological agents.

Chapter 8

Conclusion

In this project, we explore the ability of visual RL agents trained on embodied tasks to transfer visual encodings to the downstream task of depth estimation, which acts as a temporary stand-in for general perceptual abilities. We show that agents are not only able to encode knowledge of depth by simply travelling forwards in an environment, but that the depth cues generated by smooth movements are possibly the strongest determinator of an agent's transfer performance to downstream depth estimation. Collision avoidant behaviour may also leading to improved development of depth perception, but repeated tests are needed to verify this observation.

While the original goal was to investigate the contributions of embodied task variations on perceptual development, this proved to be difficult for two reasons, 1) it is not understood how relevant depth estimation is to active tasks, so our results may not generalize to evaluating with other visual metrics; and 2) many agents were not trained to full convergence, so the quality of of the learned policy at the time of evaluation became a significant confounder. Despite so, our results successfully demonstrate a rudimentary linkage between active behaviours and the subsequent development of visual skills, albeit the development not being driven by the necessity of such skills for acting.

Pretraining agents' visual encoders on depth estimation prior to RL training results in marginal effects on the training performance curves. Several agents were prone to overfitting to their training environments so the effects of depth pretraining were subdued in the evaluation environments, but it was observed to improve collision avoidance in two agents, further signifying that depth understanding is necessary for this embodied skill.

8.1 Future Directions

Here, a few possible directions are proposed:

Perceptual Metric: As mentioned in previous sections, pixelwise depth estimation should be replaced with a more sophisticated depth metric, or better, a larger suite of tasks that represent distinct perceptual base tasks. The project conducted in parallel is investigating this. Ultimately, we need to ensure that the ultimate perceptual metric is not arbitrary and can be justified as a valid measure of complex

perceptual intelligence.

Task Training: A better strategy is also needed to verify that trained agents 1) learn optimal or very good policies to maximize the quality of incoming visual cues, 2) are trained past the point of RL convergence until perceptual abilities also converge, and 3) are trained on tasks that require good visual intelligence. For example, [48] found good support of depth estimation in training *Exploration* and [11] found that their *Cache* agents outperformed navigation agents visually.

Diverse Agents: Another avenue to explore is to use pretrained RL agents from *Habitat Challenge 2020*, where research teams trained custom RL agents for tasks like *ObjectNav* and *PointNav* in the Habitat simulator. This provides a shortcut to obtaining an 'ideal' navigation agent, which is something for which we currently lack the ability to quickly train. The only caveat is that these agents receive RGBD information, which would make depth estimation somewhat redundant as the transfer task – thereby more reasons to rethink the perceptual metric. It could be possible that evaluating perceptual skills of trained agents devises an additional dimension of assessing complex visuomotor agents, which are currently evaluated by RL-based metrics like SPL.

Bibliography

[1] J. J. Gibson, *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

[2] G. Gordon, D. Kaplan, B. Lankow, D. Little, J. Sherwin, B. Suter, and L. Thaler, "Toward an integrated approach to perception and action: Conference report and future directions," *Frontiers in Systems Neuroscience*, vol. 5, p. 20, 2011.

[3] M. F. Land and D.-E. Nilsson, *Animal eyes.* Oxford University Press, 2012.

[4] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[5] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *arXiv preprint arXiv:1904.12901*, 2019.

[6] E. J. Gibson and R. D. Walk, "The "visual cliff"," *Scientific American*, vol. 202, no. 4, pp. 64–71, 1960.

[7] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[8] D. P. McGovern, B. S. Webb, and J. W. Peirce, "Transfer of perceptual learning between different visual tasks," *Journal of vision*, vol. 12, no. 11, pp. 4–4, 2012.

[9] B. S. Webb, N. W. Roach, and P. V. McGraw, "Perceptual learning in the absence of task or stimulus specificity," *PLoS One*, vol. 2, no. 12, p. e1323, 2007.

[10] I. Fine and R. A. Jacobs, "Comparing perceptual learning across tasks: A review," *Journal of vision*, vol. 2, no. 2, pp. 5–5, 2002.

[11] L. Weihs, A. Kembhavi, K. Ehsani, S. M. Pratt, W. Han, A. Herrasti, E. Kolve, D. Schwenk, R. Mottaghi, and A. Farhadi, "Learning generalizable visual representations via interactive gameplay," *arXiv preprint arXiv:1912.08195*, 2019.

[12] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[13] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV).* IEEE, 2016, pp. 239–248.

[14] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European conference on computer vision.* Springer, 2012, pp. 746–760.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[16] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International journal of computer vision*, vol. 1, no. 4, pp. 333–356, 1988.

[17] D. Terzopoulos and T. F. Rabie, "Animat vision: Active vision in artificial animals," in *Proceedings of IEEE International Conference on Computer Vision*. IEEE, 1995, pp. 801–808.

[18] G. de Croon, I. G. Sprinkhuizen-Kuyper, and E. O. Postma, "Comparing active vision models," *Image and Vision Computing*, vol. 27, no. 4, pp. 374–384, 2009.

[19] D. Floreano, T. Kato, D. Marocco, and E. Sauser, "Coevolution of active vision and feature selection," *Biological cybernetics*, vol. 90, no. 3, pp. 218–228, 2004.

[20] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[21] E. Myin and J. Degenaar, "Enactive vision," 2014.

[22] M. Suzuki and D. Floreano, "Enactive robot vision," *Adaptive Behavior*, vol. 16, no. 2-3, pp. 122–128, 2008.

[23] T. Takahashi, T. Tanaka, K. Nishida, and T. Kurita, "Self-organization of place cells and reward-based navigation for a mobile robot," in *Proceedings of the 8th International Conference on Neural Information Processing, Shanghai (China)*. Citeseer, 2001, pp. 14–18.

[24] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," *Frontiers in neurorobotics*, vol. 1, p. 3, 2007.

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

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[27] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[28] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[29] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on learning theory*. JMLR Workshop and Conference Proceedings, 2012, pp. 39–1.

[30] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[31] M. Schwarzer, N. Rajkumar, M. Noukhovitch, A. Anand, L. Charlin, R. D. Hjelm, P. Bachman, and A. C. Courville, "Pretraining representations for data-efficient reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[32] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," *arXiv preprint arXiv:2007.05929*, 2020.

[33] A. Srinivas, M. Laskin, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," *arXiv preprint arXiv:2004.04136*, 2020.

[34] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 9870–9879.

[35] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in atari," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

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

[37] G. V. Cruz Jr, Y. Du, and M. E. Taylor, "Pre-training neural networks with human demonstrations for deep reinforcement learning," *arXiv preprint arXiv:1709.04083*, 2017.

[38] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[40] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3712–3722.

[41] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey." *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[42] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?" *Science Robotics*, 2019.

[43] M. Yang and O. Nachum, "Representation matters: Offline pretraining for sequential decision making," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 784–11 794.

[44] C. W. Anderson, M. Lee, and D. L. Elliott, "Faster reinforcement learning after pretraining deep networks to predict state dynamics," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[45] W. B. Shen, D. Xu, Y. Zhu, L. J. Guibas, L. Fei-Fei, and S. Savarese, "Situational fusion of visual representation for visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2881–2890.

[46] E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Actor-mimic: Deep multitask and transfer reinforcement learning," *arXiv preprint arXiv:1511.06342*, 2015.

[47] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Reinforcement learning with prototypical representations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 920–11 931.

[48] A. Sax, B. Emi, A. R. Zamir, L. Guibas, S. Savarese, and J. Malik, "Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies," *arXiv preprint arXiv:1812.11971*, 2018.

[49] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin, "Learning to see before learning to act: Visual pre-training for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7286–7293.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[51] A. Yonas and C. E. Granrud, "The development of sensitivity of kenetic, binocular and pictorial depth information in human infants," in *Brain mechanisms and spatial vision*. Springer, 1985, pp. 113–145.

[52] J. E. Cutting and P. M. Vishton, "Perception of space and motion, chapter perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth," *Academic Pr*, vol. 46, pp. 69–117, 1995.

[53] Z. Cattaneo, T. Vecchi, C. Cornoldi, I. Mammarella, D. Bonino, E. Ricciardi, and P. Pietrini, "Imagery and spatial processes in blindness and visual impairment," *Neuroscience & Biobehavioral Reviews*, vol. 32, no. 8, pp. 1346–1360, 2008.

[54] I. P. Howard, *Perceiving in depth*. Oxford University Press, 2012.

[55] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9339–9347.

[56] J. M. Foley, "Binocular distance perception." *Psychological review*, vol. 87, no. 5, p. 411, 1980.

[57] D. Ognibene and G. Baldassare, "Ecological active vision: Four bioinspired principles to integrate bottom–up and adaptive top–down attention tested with a simple camera-arm robot," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 1, pp. 3–25, 2015.

[58] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *arXiv preprint arXiv:1301.3572*, 2013.

[59] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[60] A. Dasgupta, J. Duan, M. H. Ang Jr, Y. Lin, S.-h. Wang, R. Baillargeon, and C. Tan, "A benchmark for modeling violation-of-expectation in physical reasoning across event categories," *arXiv preprint arXiv:2111.08826*, 2021.

[61] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[62] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[66] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *arXiv preprint arXiv:2103.04918*, 2021.

[67] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[68] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[69] M. Rolinek and G. Martius, "L4: Practical loss-based stepsize adaptation for deep learning," *arXiv preprint arXiv:1802.05074*, 2018.

[70] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[71] A. S. Etienne and K. J. Jeffery, "Path integration in mammals," *Hippocampus*, vol. 14, no. 2, pp. 180–192, 2004.

[72] J. M. Loomis, R. L. Klatzky, R. G. Golledge, J. G. Cicinelli, J. W. Pellegrino, and P. A. Fry, "Nonvisual navigation by blind and sighted: assessment of path integration ability." *Journal of Experimental Psychology: General*, vol. 122, no. 1, p. 73, 1993.

[73] C. Beste and H. R. Dinse, "Learning without training," *Current Biology*, vol. 23, no. 11, pp. R489–R499, 2013.

[74] V. Leclercq and A. R. Seitz, "The impact of orienting attention in fast task-irrelevant perceptual learning," *Attention, Perception, & Psychophysics*, vol. 74, no. 4, pp. 648–660, 2012.

[75] T. Watanabe and Y. Sasaki, "Perceptual learning: toward a comprehensive theory," *Annual review of psychology*, vol. 66, pp. 197–221, 2015.

[76] D. Pascucci, T. Mastropasqua, and M. Turatto, "Monetary reward modulates task-irrelevant perceptual learning for invisible stimuli," *PLoS One*, vol. 10, no. 5, p. e0124009, 2015.

[77] M. Hoffmann and R. Pfeifer, "Robots as powerful allies for the study of embodied cognition from the bottom up," *arXiv preprint arXiv:1801.04819*, 2018.

[78] J. Z. Leibo, C. d. M. d'Autume, D. Zoran, D. Amos, C. Beattie, K. Anderson, A. G. Castañeda, M. Sanchez, S. Green, A. Gruslys *et al.*, "Psychlab: a psychology laboratory for deep reinforcement learning agents," *arXiv preprint arXiv:1801.08116*, 2018.