Curious Representation Learning for Embodied Intelligence

Yilun Du
MIT

Chuang Gan
MIT-IBM Watson AI Lab

Phillip Isola
MIT

Abstract

Self-supervised representation learning has achieved remarkable success in recent years. By subverting the need for supervised labels, such approaches are able to utilize the numerous unlabeled images that exist on the Internet and in photographic datasets. Yet to build truly intelligent agents, we must construct representation learning algorithms that can learn not only from ***datasets*** *but also learn from* ***environments****. An agent in a natural environment will not typically be fed curated data. Instead, it must explore its environment to acquire the data it will learn from. We propose a framework, curious representation learning (CRL), which jointly learns a reinforcement learning policy and a visual representation model. The policy is trained to maximize the error of the representation learner, and in doing so is incentivized to explore its environment. At the same time, the learned representation becomes stronger and stronger as the policy feeds it ever harder data to learn from. Our learned representations enable promising transfer to downstream navigation tasks, performing better than or comparably to ImageNet pretraining without using any supervision at all. In addition, despite being trained in simulation, our learned representations can obtain interpretable results on real images. Code is available at* https://yilundu.github.io/crl/.

1. Introduction

Similar to biological agents, self-supervised agents learn representations without explicit supervisory labels [38]. Impressively, these methods can surpass those based on supervised learning [12]. Yet the most successful approaches also depart from biological learning in that they depend on a curated dataset of observations to learn from.

In stark contrast, learning in biological vision involves active physical exploration of an environment. Infants are not endowed with existing visual experience, but must instead explore to obtain such experience from the surrounding environment. By playing with toys, through actions such as pushing, grasping, sucking, or prodding, infants are able to obtain experiences of texture, material, and physics [19].

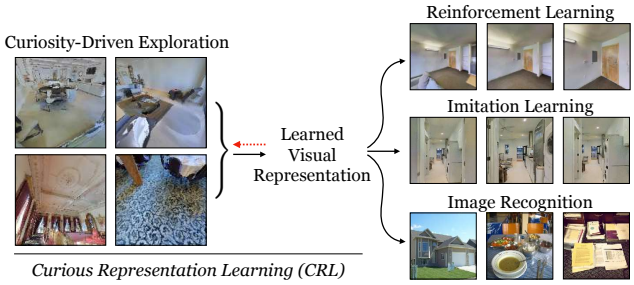

Figure 1: When put in a new world, without an explicit task or goal, we are still able to actively explore and interact with our surroundings. Our framework, CRL, enables agents to learn visual representations from interaction without any supervision, using only curiosity-driven exploration where the agent seeks out observations that incur high error under the representation model. The resultant representations enable agents to perform well in downstream reinforcement and imitation learning tasks, and further are able to transfer to recognition of real images.

By crawling into new rooms, toddlers obtain experiences of layout and geometry. This setup adds additional challenges towards learning visual representations. Algorithms must now selectively explore and determine which portion of the environment will allow for the most useful increase in visual experience. Furthermore, algorithms must also adapt to constant domain shift; at any time point, the only observed visual experience is that of a particular room, or that of a particular object being interacted with.

Given an interactive environment, and no prior data or tasks, how may we obtain a good visual representations? This is a challenging question and requires an agent to answer several subquestions. In particular, how can we learn to effectively explore and perceive the surrounding world? And, how can we integrate each different experience together to obtain the best representation possible? In this paper, we propose a unified framework towards solving these tasks.

One approach is to train a vision-based reinforcement learning agent in the interactive environment. Intuitively, as the agent learns to interact in its surrounding environment, its underlying vision system must also learn to understand the surrounding environment. A core difficulty, however, is the noisy and sparse supervision that reinforcement learning

provides, inhibiting the formation of a strong vision system. An alternative approach is thus to leverage self-supervised representation learning techniques to learn representations in embodied environments. To gather data to learn the representation, a separate exploration algorithm may be used. However, such an approach raises additional challenges. Given a new embodied environment, how can we learn to effectively explore and obtain diverse data to train our representation? And how can we continuously gather images that remain visually salient to our algorithm?

To address these issues, we propose a unified framework, Curious Representation Learning (CRL Figure 1). Our key idea is to automatically learn an exploration policy given a self-supervised representation learning technique by training a reinforcement learning (RL) to maximize reward equal to the loss of the self-supervised representation learning model. We then train our self-supervised model by minimizing the loss on the images obtained by the exploration policy. By defining the reward to our exploration policy in such a manner, it serves as a natural measure of visual novelty, as only on unfamiliar images will the loss be large. Thus, our policy learns to explore the surrounding environment and obtain images that are visually distinct from those seen in the past. Simultaneously, our self-supervised model benefits from diverse images, specifically obtained to remain visually salient to the model.

Given an embodied visual representation, we further study how it may be used for downstream interactive tasks. Interactive learning, through either reinforcement learning or behavioral cloning, is characterized by both sparse and noisy feedback. Feedback from individual actions is delayed across time and dependent on task completion, with feedback containing little information in the case a task failure, and giving conflicting results when other actions effect task completion. Such noise can quickly destroy learned visual representations. We find to enable good downstream interactive transfer, it is crucial to freeze visual network weights before transfer. We observe that our method can significantly boost the semantic navigation performance of RL policies and visual language navigation using imitation learning.

Our contributions in this paper are three-fold. First, we introduce CRL as an approach to embodied representation learning, in which a representation learning model plays a minimax game with an exploration policy. Second, we show that learned visual representations can help in a variety of embodied tasks, where it is crucial to freeze representations to enable good performance. Finally, we show that our representations, while entirely trained in simulation, can obtain interpretable results on real photographs.

2. Related Work

Self-supervised Visual Representation Learning: Unsupervised representation learning has seen increased interest in recent years [6, 18, 23, 50]. Approaches towards unsupervised learning include colorizing images [50], predicting image rotations [18] and geometry transformation [15], solving jigsaw puzzles [27], and adversarial inference [13]. Recently, approaches based on maximizing mutual information have achieved success [4, 10, 20, 22, 23, 28, 43]. While previous approaches have considered learning visual representations in static datasets, we consider the distinct problem of obtaining visual representations in interactive environments, where an agent must actively obtain data it is trained on. We provide a framework for learning a task-agnostic representation for different downstream interactive tasks.

Curiosity-based Learning: Our approach is also related to existing work in curiosity. Curiosity has also been studied extensively in the past years [5, 8, 14, 24, 29, 34, 35], as both an incentive for exploration as well as a means of achieving emergent complex behavior. Recent works have formulated curiosity as a reward dependent on a learned model, such as an inverse dynamics model [29], learned features in a VAE [7], and features from a random network [8]. In contrast to these work, which often rely on heuristic design choices to select rewards for each task, we construct curiosity as a minimax game between a generic representation learning algorithm and a reinforcement learning policy. This formulation then allows us to substitute existing representation learning algorithms into a curiosity-based formulation, enabling us to combine advances in representation learning and curiosity-based exploration.

Embodied Representation Learning: Pinto et al. [30] investigated representations that emerge from physical interaction in robotics. In contrast to our work, interactions are manually designed. Agrawal et al. [1] investigated emergent physical representations from poking in robotics. However, interactions are randomly generated and limited only to poking. Overall, our work focuses on both learning to interact and on representations in an interactive environment, and further focuses on reusing them for downstream applications.

Representation learning in RL: Recently, using unsupervised/self-supervised representation learning methods to improve sample efficiency and/or performance in RL has gained increased popularity [2, 25, 37, 39, 40]. In contrast to prior work, which focuses on synthetic game environments, we study representation learning in photorealistic 3D environments [16, 33, 48].

Perhaps most related to our work is the work of Ye [49], who show that using auxiliary tasks can improve PointGoal navigation results in the Gibson environment [47]. Different from them, we mainly investigate if we can, in a self-supervised manner, learn a generic and task-agnostic representation that can be reused for downstream interactive tasks. Concurrent to our work, Ramakrishnan *et al.* [31] also investigate learning environment-level representations through environment predictive coding and show benefits to

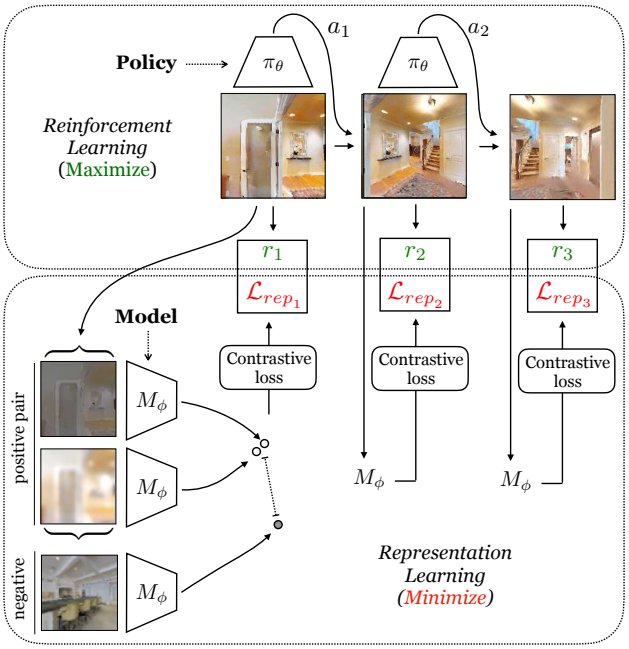

Figure 2: Overview of CRL (curious representation learning). We jointly train a RL policy and visual representation learning model to learn visual representations in interactive environments. The RL policy and visual representation learning model engage in a mini-max game where the policy maximizes reward which is set to representation learning model's loss, while the model minimizes its own loss. For the representation learning model we use the contrastive learning method SimCLR [10]. In the figure, we only diagram the full contrastive setup for the first frame, but note that it is applied to each frame.

downstream visual exploration tasks. However, it remains unclear whether these learned representations could help in more challenging navigation tasks.

3. Curious Representation Learning

Our goal is to study how to obtain a generic, task-invariant representation for downstream interactive learning in an embodied environment, without either task or extrinsic reward specification. We propose curious representation learning (CRL), a unified framework for learning visual representations. We first review some background knowledge on the contrastive representation learning framework, and next describe CRL, which extends any generic representation learning objective to interactive environments. We then describe our overall policy and model optimization procedure and evaluation protocol and provide psuedocode in the appendix.

3.1. Contrastive Representation Learning

To learn representations, we utilize contrastive learning [4, 10, 20, 28, 43, 46]. Following [10], our contrastive learning setup consists of a representation learning model M_ϕ and 2 layer MLP projection head g_ψ, and a family of data augmentations $\mathcal{T}$.

For a set of N images, $\{\boldsymbol{x}_k\}_{k=1\ldots N}$, we sample $2N$ different augmentations from $\mathcal{T}$, and apply two separate augmentations to each image to obtain pairs of augmented images $\{\tilde{\boldsymbol{x}}_k^1, \tilde{\boldsymbol{x}}_k^2\}_{k=1\ldots N}$. For a given image $\boldsymbol{x}$, we obtain a latent representation z by applying $\boldsymbol{z} = \text{Normalize}(g_\psi(M_\phi(\boldsymbol{x})))$, where we L2 normalize the output of the projection head.

We then train our contrastive loss utilizing the InfoNCE loss [28], which consists of

$$\mathcal{L}_{\text{contrast}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(\tilde{\boldsymbol{z}}_i^1, \tilde{\boldsymbol{z}}_i^2)/\tau)}{\Sigma_{j,k=1}^{N}\exp(\text{sim}(\tilde{\boldsymbol{z}}_j^1, \tilde{\boldsymbol{z}}_k^2)/\tau)} \quad (1)$$

where $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ corresponds to the dot product between latents $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. We utilize $\tau = 0.07$ and define $\mathcal{T}$ to consist of horizontal flips, random resized crops, and color saturation, using default parameters from [10].

3.2. Learning Representations from Intrinsic Motivation

When representation learning is applied to a static dataset, a model M_ϕ is trained to minimize a representation learning $\mathcal{L}_{\text{rep}}$ objective (with the objective Equation 1 corresponding to contrastive representation learning) over observed images $\boldsymbol{x}$, where images are drawn from a data distribution p_{data}

$$\min_\phi \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\mathcal{L}_{\text{rep}}(M_\phi, \boldsymbol{x})\right]. \quad (2)$$

In contrast, in our interactive setting, images in our data distribution must now be actively chosen by an agent. We utilize a reinforcement learning policy π_θ to represent our agent, where a policy is trained to maximize reward r_t at each timestep t

$$\max_\theta \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi_\theta}\left[\sum_{t=0}^{T} r_t\right]. \quad (3)$$

In our setting, we do not have access to an underlying task or reward, so we need to define implicitly define our reward. In CRL, we note that we can directly use Equation 2 to define the reward function at each time step to train our reinforcement learning policy. Specifically, we use the loss of the representation learning criterion to be our reward so that our reinforcement learning objective is now:

$$\max_\theta \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi_\theta}\left[\sum_{t=0}^{T} \mathcal{L}_{\text{rep}}(M_\phi, \boldsymbol{x})\right]. \quad (4)$$

This objective then encourages our policy to find images that M_ϕ incurs high losses on, giving a natural incentive for our policy to obtain interesting data to train our representation learning model.

Furthermore, note that while Equation 2 minimizes $\mathcal{L}_{\text{rep}}$ using M_ϕ, Equation 4 maximizes $\mathcal{L}_{\text{rep}}$ using π_θ, leading to an overall mini-max game objective

$$\max_{\theta} \min_{\phi} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \pi_\theta} \left[\sum_{t=0}^{T} \mathcal{L}_{\text{rep}}(M_\phi, \boldsymbol{x}) \right]. \tag{5}$$

This mini-max game can be seen as a synergistic way to improve both a policy and a representation learning model. This new objective encourages our policy π_θ to learn complex navigation navigation and perception patterns so that it can effectively obtain images to surprise the representation learning model M_ϕ. Simultaneously, this also allows our representation learning model to learn good representations that are resistant to samples found from the policy π_θ.

Our formulation of CRL is similar to prior work in intrinsic motivation and curiosity [29]. Such papers encourage reinforcement learning agents to explore by giving agents reward equal some predictive loss. By interpreting the representation learning loss as a predictive loss, CRL can thus be seen a curiosity model. However, while different past papers have proposed separate objectives for predictive error, such as random features [8] and inverse dynamics [29], CRL provides a generic framework to further construct different curiosity objectives, by utilizing different representation learning models, automating the traditionally hand-designed process. Furthermore, CRL allows us to reinterpret existing curiosity objectives as different methods to obtain an underlying representation of the world.

3.3. Model and Policy Optimization

When training our policy, we found that directly defining our reward following Equation 1 led to a failure case where $\mathcal{L}_{\text{contrast}}$ loss could be maximized by having an agent stand in space (as all identical image observations maximizes the denominator of $\mathcal{L}_{\text{contrast}}$). To remedy this issue, we define the reward to our policy as only be the numerator of $\mathcal{L}_{\text{contrast}}$, $r_t = -\text{sim}(\boldsymbol{x}^1, \boldsymbol{x}^2)$. We further add a constant of 1 to all rewards to ensure that rewards at each observed image is non-negative. Furthermore, following [7], we normalize rewards by the standard deviation of past observed rewards to ensure that reward magnitudes do not change significantly.

Given computed rewards r_t^γ, we use the proximal policy optimization (PPO) [36], to train our policy and optimize the objective $L(\theta) = \mathbb{E}[\min(c_t(\theta)A_t, \text{clip}(c_t(\theta), 1-\epsilon, 1+\epsilon)A_t]$, where the clip ratio $c_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and the advantage, A_t, is computed using the value function $V(s_t)$. We optimize both π_θ and M_ϕ using collected minibatches of data from PPO. Please see the appendix for pseudocode.

3.4. Experimental Protocol

Here we describe the protocols used for our empirical experiments. First, we discuss our protocol for learning embodied representations, and then our protocol for validating the utility of learned embodied representations to downstream tasks.

Representation Pretraining. To pretrain representations, we train CRL on Habitat simulator using the Matterport3D dataset [9] for 10 million interactions, reserving the Gibson dataset [47] for experimental validation. We train agents with 16 environments in parallel. Our observation space consists of only 256×256 RGB observations, and our action space consists of actions move forward by 0.25 meters, turn left by $30°$, turn right by $30°$, look up by $10°$ and look down by $10°$, with a maximum episode length of 500 steps.

Downstream Evaluation. We evaluate pretrained representations on the downstream tasks of semantic navigation, visual language navigation, and real image understanding. For semantic navigation, we train a reinforcement learning agent using the Habitat simulator on the Gibson dataset and on object navigation using the Habitat Matterport3D dataset (due to the lack of object annotations in Gibson). For visual language navigation, we utilize an imitation learning agent on Matterport3D dataset, and on real images we utilize the Places dataset. We use the default environment settings for both in Habitat [33]. We utilize the *same* representation across different tasks using the features from the last final average pooling layer of a ResNet50. To enable effective interactive downstream transfer, we found that it was *crucial* to freeze visual representations, due to the noisy nature of gradients from interactive tasks. Such a technique has also been noted to be useful in few-shot learning [44].

Model Architectures. For representation learning models M_ϕ and policies π_θ, we utilize a ResNet50 [21] image encoder. To enable stable reinforcement learning, we replace batch normalization layers with group normalization. To train M_ϕ, we use a 2 layer projection head, with a projection dimension of 128 dimensions.

4. Experiments

We quantitatively and qualitatively show that CRL can learn generic, task-agnostic visual representations for downstream interactive tasks. We discuss our experimental setup in Section 4.1. Next we analyze the interactive behavior of CRL in Section 4.2. Using one unified pretrained model, we show that we can improve performance in semantic navigation in RL in Section 4.3, visual-language navigation using imitation learning in Section 4.4, and can further achieve transfer to recognition of real images in Section 4.5. Finally, we discuss how we may also obtain representations in a real biological setting in the appendix.

4.1. Experimental Setup

To pretrain representations, we train different models on the Habitat simulator using the Matterport3D dataset for 10

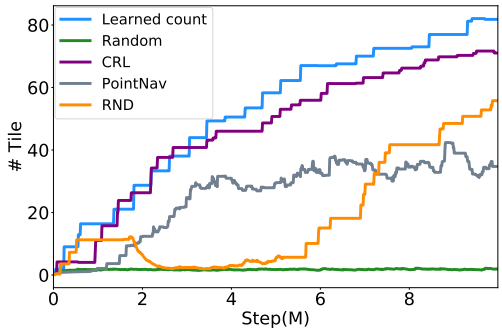

Figure 3: Plots of the average number of tiles explored in a across separate environments of different reinforcement learning agents. To gather images that have high contrastive loss, CRL explores effectively around the surrounding environment, outperforming RND and PointNav agents, and performs similar to a learned counts agent that is explicitly encouraged to maximize tile exploration.

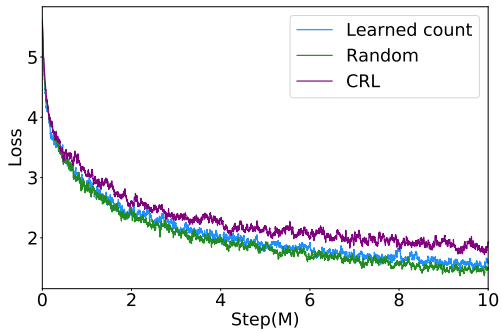

Figure 4: Plots of contrastive loss over time using different exploration methods. By treating the process of image gathering as an adversarial process, CRL enables the procurement of diverse images, leading to larger contrastive loss.

Random Exploration (LPIPS Diversity 0.708) | Learned Counts (LPIPS Diversity 0.717) | CRL (LPIPS Diversity 0.728)

Figure 5: Illustration of data acquired for contrastive training utilizing either a random agent, a learned counts agent or CRL on 4 different environments. Data collected by a random agent exhibit limited diversity, while data collected by a learned counts are diverse but not visually interesting (indicated by black backgrounds). Data collected by CRL is diverse.

million interactions. In addition to CRL, we consider the following set of baseline methods to obtain representations:

Exploration Strategies. In CRL, we rely on an intrinsically motivated policy to explore the surrounding world to train our contrastive model. We further compare using other approaches to collect data from the surrounding environment. We consider either using random actions to explore or using a learned counts-based exploration approach [42].

Video Game Methods. Concurrent to our work, recent work has explored learning state representation for reinforcement learning in static video game environments. These works assume the presence of a statically collected dataset of experiences, and are customized to non-realistic video game settings. We compare with one such recent approach, augmented temporal constrast (ATC) [40], where we utilize the exploration policy of CRL to explore the surrounding environment, and utilize ATC to learn a representation on top of the gathered images.

Curiosity Objective. Under CRL, we may interpret other existing curiosity-based reinforcement learning approaches as representation learning objectives. We thus compare with one such objective, that of Random Network Distillation (RND) [8], which trains a model to regress the representations of a frozen network. We use RND to both incentive exploration, as well as learn a representation of the scene.

Policy Based Representation. An alternative approach to obtain representations is to utilize a reinforcement learning policy to learn representations of the environment. We thus compare with the representation learned by a PointNav policy trained on the Matterport training split.

ImageNet Pretraining. We may also use existing large scale vision datasets to obtain our model. We thus provide a comparison study where we initialize our model using a ResNet50 pretrained on ImageNet.

All models are trained in Pytorch using PPO for 10M frames with the Adam optimizer. Hyperparameters for both representation pretraining and downstream evaluation are provided in the supplement.

4.2. Visual Exploration

We first assess the ability of each method to actively explore the environment around it. We report the average number of 0.01×0.01 tiles explored (as measured by x, y position in the simulator) in a given scene over the course of training in Figure 3. We find that CRL learns to explore well, outperforming both random policies and our curiosity baseline (RND) as well a PointNav policy trained explicitly to navigate around the environment. We find that CRL performs similarly in terms of exploration to a learned counts-based exploration approach, but note that the learned counts-based exploration is explicitly encouraged to explore the surrounding environment, maintaining a count of explored tiles using a learned hash map, while CRL encourages the policy to gather diverse data for representation learning.

We next evaluate the ability of CRL to gather diverse data to train our contrastive model. We compare utilizing either random exploration, a learned counts-based method, or CRL to obtain data to train a contrastive model. In Figure 4, we plot contrastive loss curves obtained by utilizing data collected by each method. We find that since CRL is trained

Table 1: Full results of comparisons of embodied navigation with learned interactive representations average across 5 seperate seeds (with standard error in parentheses). Policies are evaluated on the test set of ImageNav and ObjectNav tasks and are trained for 10 million frames in each environment. We report the mean across 3 different seeds and report results of individual runs in the supplement. We consider either training an RL agent from scratch, utilizing existing representation learning methods (ATC [40], RND [8] and contrastive learning) or utilizing supervised weights (PointNav Policy, ImageNet Initialization). RL agents initialized from pretrained weights have representations frozen, while all weights in the from scratch RL agent are trained.

Environment	Category	Method	SPL↑	Soft SPL↑	Success↑	Goal Distance↓
ImageNav	From Scratch		0.0207 (0.0012)	0.173 (0.007)	0.039 (0.003)	4.85 (0.04)
	Other Representation Learning Algorithms	RND [8]	0.0158 (0.0027)	0.124 (0.013)	0.029 (0.004)	5.29 (0.08)
		ATC [40]	0.0268 (0.0029)	0.172 (0.013)	0.059 (0.004)	4.63 (0.04)
	Contrastive Learning	Random Exploration	0.0285 (0.0014)	0.195 (0.010)	0.054 (0.003)	4.68 (0.04)
		Learned Counts [42]	0.0277 (0.0030)	0.183 (0.011)	0.057 (0.003)	**4.54 (0.08)**
		CRL (ours)	**0.0324 (0.0018)**	**0.219 (0.005)**	**0.058 (0.002)**	4.55 (0.04)
	Supervised	PointNav Policy	0.0254 (0.0021)	0.187 (0.020)	0.048 (0.002)	4.66 (0.03)
		ImageNet Initialization	0.0193 (0.0042)	0.143 (0.022)	0.050 (0.007)	4.61 (0.01)
ObjectNav	From Scratch		0.0010 (0.0006)	0.037 (0.008)	0.003 (0.002)	7.94 (0.44)
	Other Representation Learning Algorithms	RND [8]	0.0000 (0.0000)	0.007 (0.001)	0.000 (0.000)	7.96 (0.13)
		ATC [40]	0.0020 (0.0014)	0.058 (0.013)	0.003 (0.002)	8.32 (0.27)
	Contrastive Learning	Random Exploration	0.0042 (0.0007)	0.076 (0.010)	0.011 (0.002)	7.39 (0.19)
		Learned Counts [42]	0.0079 (0.0013)	0.110 (0.008)	0.026 (0.004)	7.49 (0.17)
		CRL (ours)	**0.0144 (0.0046)**	**0.119 (0.007)**	**0.040 (0.019)**	7.33 (0.13)
	Supervised	PointNav Policy	0.0390 (0.0011)	0.094 (0.005)	0.007 (0.002)	**7.29 (0.08)**
		ImageNet Initialization	0.0064 (0.0021)	0.062 (0.004)	0.010 (0.003)	7.91 (0.10)

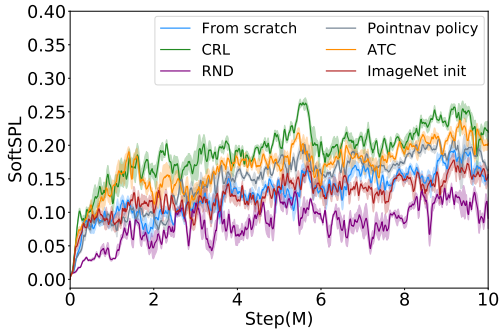

Figure 6: Plot of training SoftSPL over training steps of CRL compared to other visual representation learning methods across 5 separate seeds in reinforcement learning. CRL performs significantly better than initialization from scratch and outperforms all other methods on ImageNav in Gibson.

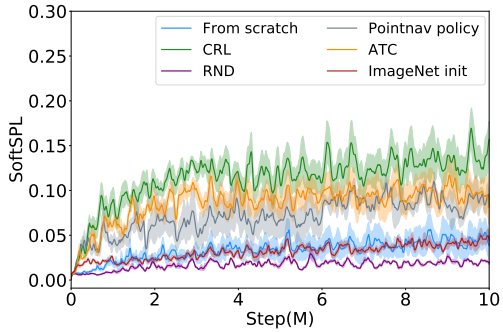

Figure 7: Plot of SoftSPL over training steps of CRL compared to other visual representation learning methods across 5 separate seeds in reinforcement learning. CRL performs significantly better than initialization from scratch and outperforms all other methods on ObjectNav in Matterport3D.

to adversarialy generate data for the contrastive model, the overall contrastive loss is significantly higher in later stages of training. We further visualize batches of image data collected from different methods in Figure 5, observing a high degree of visual diversity in the data collected through CRL. We quantitatively observe larger diversity utilizing the LPIPS diversity metrics [51] with details in Section **??**.

4.3. Semantic Navigation with RL

Next, we investigate how each of the learned repreesentations Section 4.1 can be utilized to learn effective RL policies for semantic navigation in the data-efficient setting.

Setup. We measure reinforcement learning using the standard ImageNav task on the Gibson environment, and ObjectNav task on the Matterport3D environment included in Habitat [33]. Since we aim to validate the efficacy of learned visual representations, we train reinforcement learning policies using only 256×256 RGB inputs, processed using learned representations, for 10 million frames. This setting is thus more challenging setting than typically evaluated in [33] as we assume the absence of either depth or robot localization information which is often given.

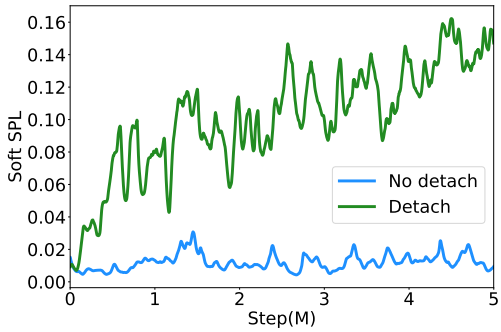

Figure 8: Plot of SoftSPL of a reinforcement learning agent on ObjectNav when the visual representation of CRL is frozen or not frozen. Due to noisy gradient updates, when weights of CRL are not frozen, performance deteriorates significantly.

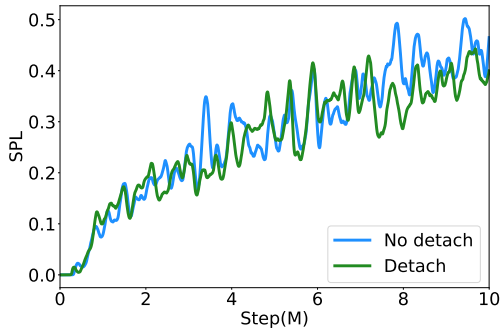

Figure 9: Plot of SPL of a reinforcement learning agent on PointNav, when convolution weights of a *random* network are either frozen or not frozen. Surprisingly, we find that in PointNav, there is *no* difference between each setting (we observe large differences in both ObjectNav and ImageNav tasks). Our results indicate that visual representations may not be effective enough in the PointNav task to show performance gains in the limited data regime we study.

Metrics. We report standard metrics of visual navigation. We report tasks success, success weighted by path length (SPL) [3], soft SPL (success weighted by path length [11], but with a softer success criterion), and distance to goal. We utilize default criteria defined in [33].

Baselines. We compare learning policies with representations from each approach described in Section 4.1. We consider utilizing representations generated via different exploration strategies (Random, Learned Counts), using video game methods (ATC), changing the underlying curiosity objective (RND), extracted by an RL policy (PointNav), and weights initialized from ImageNet. In each setting, we freeze convolutional weights, which we find crucial for good performance. We further compare with training an RL policy trained end-to-end entirely from scratch (From Scratch), including convolutional weights. We found that freezing weights for the From Scratch policy significantly dropped performance (a fall of over 0.07 SoftSPL).

Results. We run each separate representation learning approach across 3 different random seeds, with mean performance across each metric reported in Table 1. Following [3], we recommend primarily looking at SPL and SoftSPL as metrics of performance. On both ImageNav and ObjectNav, we find that CRL performs the best. Overall, we find that representations learned through contrastive learning lead to the best reinforcement learning performance. Subsequently, we find that weights from either PointNav weights or ATC perform better than training a policy from scratch. Surprisingly, we find that utilizing ImageNet does not appear to improve reinforcement learning performance significantly.

We visualize training SoftSPL across ImageNav in Figure 6 and across ObjectNav in Figure 7. Similar to our reported metrics, we find that CRL leads to the highest early increases in SoftSPL. Following that, we find that using weights from a trained PointNav policy or an ImageNet pretrained policy gives boosts to performance compared to a randomly initialized policy.

We note that although our result values are low, they are in line with those found in the 2020 Habitat navigation challenge*. Furthermore, we note that we study a harder version of both semantic navigation tasks – we only utilize RGB inputs to train our reinforcement learning policy, and do not assume information about either ground truth depth or the current localization of the policy.

Importance of Detaching Gradients. We further ablate the effect of freezing representations in Figure 8. We consider training a reinforcement learning policy on the object navigation task with or without freezing the weights of a reinforcement learning policy. We find that without freezing the weights of the convolutional network, SoftSPL increases significantly slower.

What Doesn't Representation Learning Help On? The most common evaluation task in Habitat [33] is the PointNav navigation task with a compass, where an embodied policy is instructed to navigate to a specific positional offset. Surprisingly, we find that learning a representation is not important in PointNav. In particular, in Figure 9, we initialize two separate policies from scratch and freeze the weights of the convolutional encoder of one policy. In both settings, we find that the overall PointNav SPL performance is *identical*. We posit that in PointNav, in our data-efficient experimental setting, vision is not crucial to obtain good performance, since the policy is given a compass, but note that in the large-scale RL setting [45] shows that vision is indeed helpful for navigation.

4.4. Instruction Navigation with Imitation Learning

We next investigate how the different representation learning methods in Section 4.1 can be utilized to aid visual language navigation (VLN) via imitation learning.

Setup. We evaluate imitation learning using the vision language instruction benchmark introduced in [26]. For sim-

*https://aihabitat.org/challenge/2020/

Table 2: Comparison of performance of each pretrained representation on instruction following evaluated in unseen validation rooms.

Setting	Method	SPL↑	Success↑	Goal Distance↓
Behavioral Cloning	From Scratch	0.138	0.152	9.17
	RND [8]	0.141	0.149	9.12
	ATC [40]	0.147	0.156	9.06
	CRL (ours)	**0.157**	**0.169**	**8.77**
	Imagenet	0.152	0.164	8.91
Dagger	From Scratch	0.192	0.206	8.32
	RND [8]	0.187	0.200	8.23
	ATC [40]	0.192	0.205	7.99
	CRL (ours)	**0.199**	**0.218**	**8.21**
	Imagenet	0.206	0.222	8.07
Random Agent	-	0.0	0.0	10.23

plicity, we utilize the base model and loss setting in [26], corresponding to training a Seq2Seq agent [41] with or without Dagger [32]. We utilize the author's implementation.

Metrics. We use the same set of metrics defined in Section 4.3. We report SPL, Success and goal distance on the val-unseen split in [26], corresponding to unseen rooms, and report results on val-seen setting in the appendix.

Baselines. We compare representations learned from CRL to those learned using either ATC or RND. We further compare representations from CRL to utilizing weights from a supervised ImageNet model.

Results. We compare each learned representation when applied to imitation learning in Table 2. In both the behavioral cloning and Dagger settings, we find that utilizing CRL obtains better performance than utilizing either random, RND, or ATC weights. We further find that CRL obtains comparable performance to the Imagenet supervised model.

4.5. Transfer to Real Image Recognition

Finally, we investigate to what extent our learned embodied representations, despite being learned entirely in simulation, can actually *transfer* to real photographic scenes.

Setup. To assess how representations transfer to realistic images, we utilize the Places dataset. We chose a subset of 59 class categories in Places corresponding to indoor room scenes (with selected class categories in the appendix). Following [50], we then measure representations by fine-tuning a linear classifier on the final averaged-pooled features of our trained ResNet50 models.

Baselines. We compare with the same set of baselines as in Section 4.4. For methods in which both an RL policy and model is learned, we evaluate the representations of both. To assess representation learning of RL policies, we also compare with the representations learned from a PointNav policy trained on the Habitat Matterport3D dataset.

Results. We report quantitative results from linear fine-tuning in Table 3. Overall, we find that CRL learns representations that transfer best to real images, outperforming other approaches. Of our remaining methods, we observe

Table 3: Comparison of pretrained embodied representations in Habitat when transferred to the Places Dataset.

Learning Objective	Representation Accuracy			
	Policy Accuracy		Model Accuracy	
	Top 1	Top 5	Top 1	Top 5
Random Initialization	-	-	9.22	27.59
RND	2.61	10.13	5.98	18.03
ATC	-	-	14.83	40.61
CRL (ours)	**4.68**	**18.32**	**21.22**	**48.78**
PointNav	4.31	15.27	-	-
ImageNet Pretraining	-	-	54.59	85.15

Figure 10: Comparative illustration of representation space nearest neighbors of CRL and a random network on room scenes in Places.

that ATC learns the second best representation. We further find that image encoders of policies learn poor representations that do not transfer well to real images, with the visual encoder of the CRL policy learning the best representation.

While our results are worse than those of an ImageNet supervised model, we emphasize that this is still strong performance on our task since our approach is trained *entirely* in simulation without any *supervision*. Qualitatively, we visualize representations from CRL by finding the nearest neighbors, in learned representation space, of different images in the Places dataset in Figure 10. Compared to a random network, we find more visually similar neighbors.

5. Conclusion

In this paper, we proposed a generic framework to learn task-agnostic visual representations in embodied environments. Our learned representations enable promising transfer on downstream semantic and language guided navigation tasks, and further can transfer to visual recognition of real photos. We hope our proposed framework inspires future work towards learning both better task-agnostic representations and transferring to more complex embodied tasks [17].

Acknowledgments. We thank MIT-IBM for support that led to this project. Yilun Du is funded by an NSF graduate research fellowship. We thank Dhruv Batra for giving helpful comments on the manuscript.

References

[1] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *NIPS*, 2016. 2

[2] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari, 2020. 2

[3] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents, 2018. 7

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019. 2, 3

[5] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016. 2

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013. 2

[7] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018. 2, 4

[8] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018. 2, 4, 5, 6, 8

[9] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 4

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3

[11] Samyak Datta, Oleksandr Maksymets, Judy Hoffman, Stefan Lee, Dhruv Batra, and Devi Parikh. Integrating egocentric localization for more realistic point-goal navigation agents. *arXiv preprint arXiv:2009.03231*, 2020. 7

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[13] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2

[14] Chuang Gan, Xiaoyu Chen, Phillip Isola, Antonio Torralba, and Joshua B Tenenbaum. Noisy agents: Self-supervised exploration by predicting auditory events. *arXiv preprint arXiv:2007.13729*, 2020. 2

[15] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*, pages 5589–5597, 2018. 2

[16] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020. 2

[17] Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, et al. The threedworld transport challenge: A visually guided task-and-motion planning benchmark for physically realistic embodied ai. *arXiv preprint arXiv:2103.14025*, 2021. 8

[18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[19] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co, 1999. 1

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 4

[22] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2

[23] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[24] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016. 2

[25] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver,

and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks, 2016. 2

[26] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments, 2020. 7, 8

[27] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[29] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017. 2, 4

[30] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *ECCV*, 2016. 2

[31] Santhosh K Ramakrishnan, Tushar Nagarajan, Ziad Al-Halah, and Kristen Grauman. Environment predictive coding for embodied agents. *arXiv preprint arXiv:2102.02337*, 2021. 2

[32] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010. 8

[33] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research, 2019. 2, 4, 6, 7

[34] Jürgen Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Institut für Informatik, Technische Universität München. Technical Report FKI-126*, 90, 1990. 2

[35] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991. 2

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017. 4

[37] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations, 2020. 2

[38] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artif. Life*, 11(1-2):13–29, 2005. 1

[39] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning, 2020. 2

[40] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning, 2020. 2, 5, 6, 8

[41] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 8

[42] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 5, 6

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2, 3

[44] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need?, 2020. 4

[45] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019. 7

[46] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[47] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 2, 4

[48] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 2

[49] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. Auxiliary tasks speed up learning pointgoal navigation. *CORL*, 2020. 2

[50] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2, 8

[51] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation, 2018. 6