Mid-Level Visual Representations Improve Generalization and Sample Efficiency for Learning Visuomotor Policies

Alexander Sax[1,2] Bradley Emi[3] Amir Zamir[1,3] Leonidas Guibas[2,3] Silvio Savarese[3] Jitendra Malik[1,2]

[1] University of California, Berkeley [2] Facebook AI Research [3] Stanford University

http://perceptual.actor/

Abstract

*How much does having **visual priors about the world** (e.g. the fact that the world is 3D) assist in learning to perform **downstream motor tasks** (e.g. delivering a package)? We study this question by integrating a generic perceptual skill set (e.g. a distance estimator, an edge detector, etc.) within a reinforcement learning framework—see Fig. 1. This skill set (hereafter **mid-level perception**) provides the policy with a more processed state of the world compared to raw images.*

*We find that using a mid-level perception confers significant advantages over training end-to-end from scratch (i.e. not leveraging priors) in navigation-oriented tasks. Agents are able to generalize to situations where the from-scratch approach fails and training becomes significantly more sample efficient. However, we show that realizing these gains requires careful selection of the mid-level perceptual skills. Therefore, we refine our findings into an efficient **max-coverage feature set** that can be adopted in lieu of raw images. We perform our study in completely separate buildings for training and testing and compare against state-of-the-art feature learning methods and visually blind baseline policies.*

1. Introduction

The renaissance of deep reinforcement learning (RL) started with the Atari DQN paper in which Mnih et al. [51] demonstrated an RL agent that learned to play video games directly from pixels. Levine et al. [45] adapted this approach to robotics by using RL for control from raw images—a technique commonly referred to as *pixel-to-torque*. The premise of direct-from-pixel learning poses a number of fundamental questions for computer vision: *are perceptual priors about the world actually necessary for learning to perform robotic tasks?* and *what is the value of computer vision objectives, if all one needs from images can be learned from scratch using raw pixels by RL?*

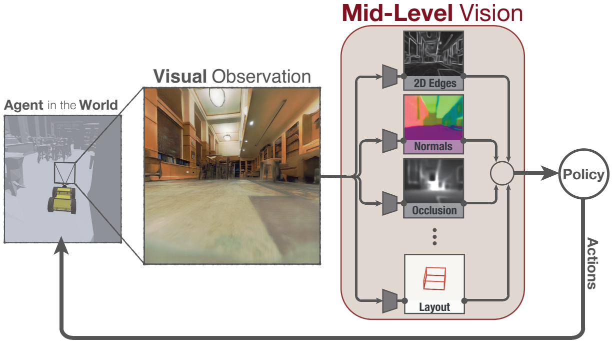

Figure 1. **A mid-level perception in an end-to-end framework for learning active robotic tasks.** We systematically study if/how a set of generic mid-level vision features can help with learning downstream active tasks. Not incorporating such mid-level perception (i.e. bypassing the red box) is equivalent to learning directly from raw pixels. We report significant advantages in *sample efficiency* and *generalization* when using mid-level perception.

While deep RL from pixels can learn arbitrary policies in an elegant, end-to-end fashion, there are two phenomena endemic to this paradigm: **I.** learning requires massive amounts of data (large sample complexity), and **II.** the resulting policies exhibit difficulties reproducing across environments with even modest visual differences (difficulty with generalization). These two phenomena are characteristic of a type of learning that is *overly generic*—in that it does not make use of available *valid assumptions*. Some examples of valid assumptions include that the world is spatially 3D or that certain groupings ("objects") behave together as a single entity. These are *facts* about the world and are generally true. Incorporating them as priors could provide an advantage over the assumption-free style of learning that always recovers the correct function when given infinite data but struggles when given a limited number of samples [24].

In this paper, we show that including appropriate perceptual priors can alleviate these two phenomena, improving *generalization* and *sample efficiency*. The goal of these priors (more broadly, one of the primary goals of perception) is to provide an internal state that is an understandable representation of the world. In conventional computer vision, this involves defining a set of offline proxy problems (e.g. object detection, depth estimation, etc.) and solving them independently of any ultimate downstream active task [11, 4]. We study how such standard mid-level vision tasks [59] and their associated features can be used with RL frameworks in order to train effective visuomotor policies.

We distill our analysis into three questions: whether these features could improve the: **I.** *learning speed* (answer: *yes*), **II.** *generalization to unseen test spaces* (answer: *yes*), and then **III.** *whether a fixed feature could suffice* or a set of features is required for supporting arbitrary motor tasks (answer: *a set is essential*).

Our findings assert that supporting downstream tasks requires *a set* of visual features. Smaller sets are desirable for both computational efficiency and practical data collection. We put forth a simple and practical solver that takes a large set of features and outputs a smaller feature subset that minimizes the worst-case distance between the selected subset and the best-possible choice. The module can be adopted in lieu of raw pixels to gain the advantages of mid-level vision.

This approach introduces visual biases in a completely computational manner, and it is intermediate between ones that learn everything (like *pixel-to-torque*) and those that leverage fixed models (like classical robotics [69]). Our study requires learning a visuomotor controller for which we adopted RL—however any of the common alternatives such as control theoretic methods would be viable choices as well. In our experiments we use neural networks from existing vision techniques [90, 93, 12, 86], trained on real images for specific mid-level tasks. We use their internal representations as the observation provided to the RL policy—we do not use synthetic data to train the visual estimators and do not assume they are perfect. When appropriate, we use statistical tests to answer our questions.

An interactive tool for comparing any trained policies with videos and reward curves, trained models, and the code are available at at `http://perceptual.actor`.

2. Related Work

Our study has connections to a broad set of topics, including lifelong learning, un/self supervised learning, transfer learning, reinforcement and imitation learning, control theory, active vision and several others. We overview the most relevant ones within constraints of space.

Offline Computer Vision encompasses the approaches designed to solve various stand-alone vision tasks, e.g. depth estimation [18, 43], object classification [42, 32], detection [64, 25], segmentation [70, 30, 35], pose estimation [92, 9, 87], etc. The approaches use various levels of supervision [42, 55, 14, 7], but the common characteristic shared across these methods is that they are *offline* (i.e. trained and tested on prerecorded datasets) and evaluated as a fixed pattern recognition problem. We study how such methods can be plugged into a larger framework for solving downstream active tasks.

Reinforcement Learning, [76, 50, 71, 45] and its variants like Meta-RL [20, 27, 54, 21, 40, 74, 17, 49] or its sister fields such as imitation learning [1, 26, 22, 89, 37, 63], commonly focus on the last part of the end-to-end active task pipeline: how to choose an action given a "state" from the world. Improvements commonly target the learning algorithm itself (e.g. PPO [67], Q-Learning [51], SAC [29], et cetera), how to efficiently explore the state space [23, 58], or how to balance exploration and exploitation [3, 5]. These can be seen as users of our method as we essentially update the input state from pixels (or single fixed features) to a set of generic and presumably more effective vision features.

Representation/Feature Learning literature shares its goal with our study: how to encode images in a way that provides benefits over using just raw pixels. There has been a remarkable amount of work in this area. They leverage the data either by making some task-agnostic assumption about the data distribution being simpler than the raw pixels (unsupervised approaches like the autoencoder family [34, 83, 41, 48] and Generative Adversarial Networks [77, 23, 57, 15]) or they exploit some known structure (so-called *self-supervised* approaches [91, 56, 55, 72, 84, 52, 28, 74, 13, 61]). A common form of self-supervision in active contexts leverages temporal information to predict unseen observations [16, 58, 36, 82], or actions [94, 2, 58, 62]. Domain adaptive approaches learn features that are task-relevant but domain agnostic [79, 65, 81, 75, 47, 60, 80, 6]. We show the appropriate choice of feature depends fiercely on the final task. Solving multiple active tasks therefore requires *a set* of features, consistent with recent works in computer vision showing a no single visual feature is the best transfer source for all vision tasks [90].

Robot Learning includes methods that leverage fixed models and make hard choices (e.g. which objects are where) [69] or model-free methods that learn everything from scratch [46]. These approaches either make assumptions about the world or else require enormous amounts of data, thus they typically work best when restricted to simple domains unrepresentative of the real world (e.g. fixed tabletop domains). For single tasks, methods that leverage some task-specific knowledge in learning have been shown to improve performance [10, 88, 38] in realistic environments.

Cognitive Psychology studies [44, 73] suggest that one mechanism for the flexible and sample efficient learning of biological organisms is a universal and evolutionarily ancient set of perceptual biases (such as an object-centric world structure) that tilt learning towards useful visual abstractions. In this paper, we are interesting in embedding a set of perceptual biases into an active artificial agent via a dictionary of mid-level visual features.

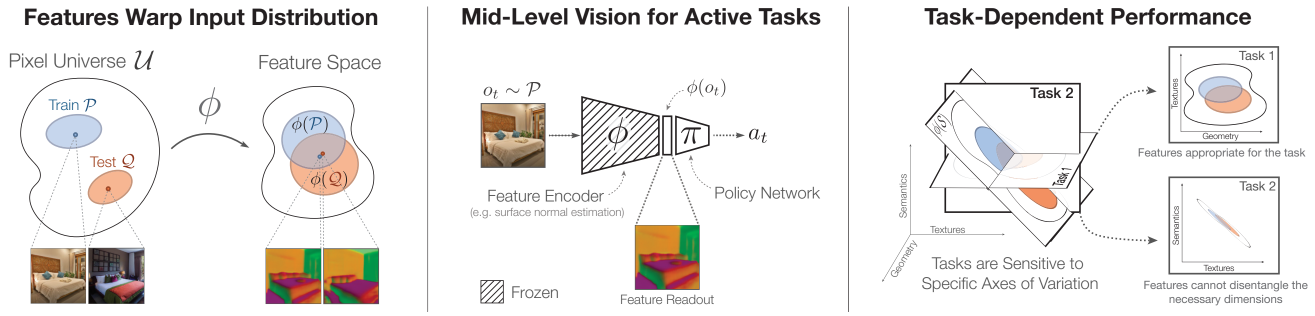
Features Warp Input Distribution
Pixel Universe $\mathcal{U}$
Feature Space
Train $\mathcal{P}$
Test $\mathcal{Q}$
ϕ
$\phi(\mathcal{P})$
$\phi(\mathcal{Q})$
Mid-Level Vision for Active Tasks
$o_t \sim \mathcal{P}$
$\phi(o_t)$
ϕ
π
a_t
Feature Encoder
(e.g. surface normal estimation)
Policy Network
Frozen
Feature Readout
Task-Dependent Performance
Task 2
Task 1
$\phi(\mathcal{S})$
Semantics
Textures
Geometry
Tasks are Sensitive to Specific Axes of Variation
Features appropriate for the task
Features cannot disentangle the necessary dimensions

Figure 2. **Illustration of our approach.** Left: Features warp the input distribution, potentially making the train and test distributions look more similar to the agent. Middle: The learned features from fixed encoder networks are used as the state for training policies in RL. Right: Downstream tasks prefer features which contain enough information to solve the task while remaining invariant to the changes in the input which are irrelevant for solving the tasks.

3. Methodology

Our study is focused on agents that maximize the reward in unknown test settings. Our setup assumes access to a set of features $\Phi = \{\phi_1, \dots, \phi_m\}$, where each feature is a function that can be applied to raw sensory data: transforming the training distribution ($\mathcal{P}$) into $\mathcal{P}_\phi \triangleq \phi(\mathcal{P})$ and transforming the test distribution ($\mathcal{Q}$) into $\mathcal{Q}_\phi \triangleq \phi(\mathcal{Q})$ (as in Fig. 2, left). We examine whether this set can be used to improve this test reward, both in terms of learning speed and generalization (*hypotheses I* and *II* in Sec. 3.2).

Visual Features: Figure 2 shows how a proper feature transforms the training distribution $\mathcal{P}$ into a feature space $\mathcal{P}_\phi$ so that the states at test-time appear similar to those seen during training. In this way, using RL to maximize the training reward also improves the test-time performance, $R_{\mathcal{Q}_\phi}$. Although there are ways [78] to bound the test performance in terms of the training performance and the shift between the two distributions $\mathcal{P}_\phi$ and $\mathcal{Q}_\phi$, these bounds are loose in practice and the question of when a feature is helpful remains an empirical one.

3.1. Using Mid-Level Vision for Active Tasks

How might we use mid-level perception to support a downstream task? Our mid-level features come from a set of neural networks that were each trained, offline, for a specific mid-level visual task (precisely, 20 networks from [90] – see Fig. 3). We freeze each encoder's weights and use the network (ϕ) to transform each observed image o_t into a summary statistic $\phi(o_t)$ that we feed to the agent. During training, only the agent policy is updated (as shown in Fig. 2, center). Freezing the encoder networks has the advantage that we can reuse the same features for new active tasks without degrading the performance of already-learned policies.

3.2. Core Questions

The following section details our three core hypotheses relating features to agent performance.

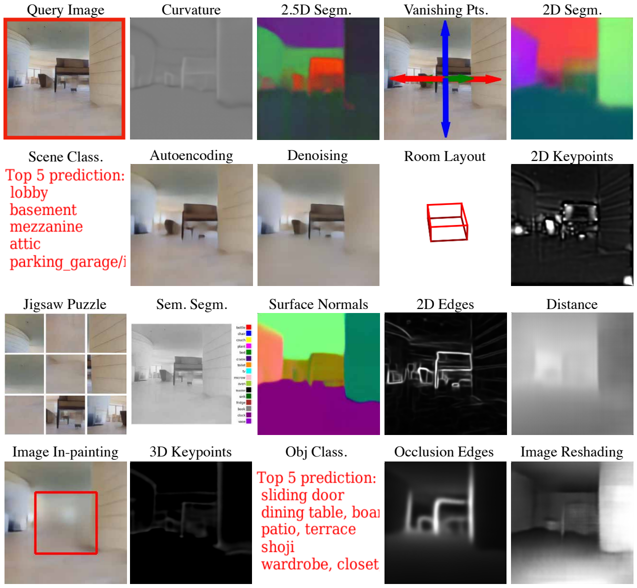

Figure 3. **Mid-level vision tasks.** Sample outputs from the vision networks (from Taskonomy [90] tested on an input from Gibson environment [86]). See more frame-by-frame results on the website.

Hypothesis I: Sample Efficiency: *Does mid-level vision provide an advantage in terms of sample efficiency when learning an active task?* We examine whether an agent equipped with mid-level vision can learn faster than a comparable agent with vision but no visual priors about the world—in other words, an agent learning *tabula rasa*.

Hypothesis II: Generalization: *Can agents using mid-level vision generalize better to unseen spaces?* If mid-level perception transforms images into standardized encodings that are less environment-specific (Fig. 2, left), then we should expect that *agents* using these encodings will learn policies that are more robust to differences between the training and testing environments. We evaluate this in HII by testing which (if any) of the m feature-based agents outperform *tabula rasa* learning in unseen test environments:

$$\bigvee_{i=1}^{m} (R_{\mathcal{Q}_{\phi_i}} > R_{\mathcal{Q}}),$$

and correcting for multiple hypothesis testing.

Hypothesis III: Single Feature or Feature Set: *Can a **single** feature support all downstream tasks? Or is a set of features required for gaining the feature benefits on arbitrary active tasks?* We demonstrate that no feature is universal and can outperform all other features regardless of the downstream activity (represented in the right subplot of Fig. 2). We show this by demonstrating cases of *rank-reversal*—when the ideal features for one task are non-ideal for another task (and vice-versa):

$$\left(R^{T}_{\mathcal{Q}_{\phi}} > R^{T}_{\mathcal{Q}_{\phi'}}\right) \wedge \left(R^{T'}_{\mathcal{Q}_{\phi'}} > R^{T'}_{\mathcal{Q}_{\phi}}\right),$$

for tasks T and T' with best features ϕ and ϕ', respectively.

For instance, we find with high confidence that *depth estimation* features perform well for visual exploration and *object classification* for target-driven navigation, but neither do well vice-versa.

3.3. A Covering Set for Mid-Level Perception

Employing a larger feature set maximizes the change of having the feature proper for the downstream task available. However, a compact set is desirable since agents using a larger set need more data to train—for the same reason that training from raw pixels requires many samples. Therefore, we propose a **Max-Coverage Feature Selector** that curates a compact subset of features to ensure the *ideal* feature (encoder choice) is never too far away from one in the set.

The question now becomes how to find the best compact set, shown in Figure 4. With a measure of distance between features, we can explicitly minimize the worst-case distance between the best feature and our selected subset (the *perceptual risk* by finding a subset $X_\delta \subseteq \Phi = \{\phi_1, ..., \phi_m\}$ of size $|X_\delta| \leq k$ that is a δ-cover of Φ with the smallest possible δ. This is illustrated with a set of size 7 in Figure 4.

The task taxonomy method [90] defines exactly such a distance: a measure between perceptual tasks. Moreover, this measure is predictive of (indeed, derived from) transfer performance. Using this distance, minimizing worst-case transfer (*perceptual risk*) can be formulated as a sequence of Boolean Integer Programs (BIPs)[1] parameterized by a boolean vector x indicating which features should be included in the set.

$$\begin{aligned} &\text{minimize: } \mathbb{1}^T x, \\ &\text{subject to: } Ax \succeq \delta \text{ and } x \in \{0,1\}^m. \end{aligned}$$

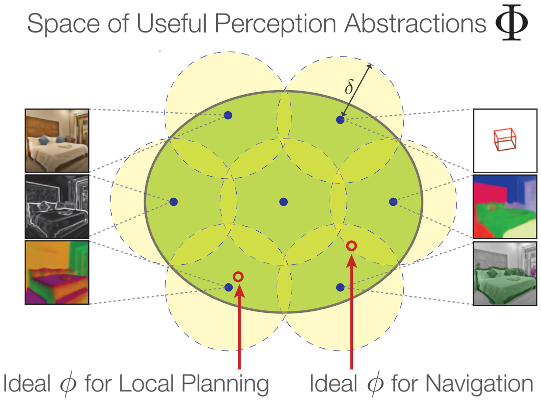
Space of Useful Perception Abstractions Φ
δ
Ideal ϕ for Local Planning
Ideal ϕ for Navigation

Figure 4. **Geometry of the feature set.** We select a covering set of features that minimizes the worst-case distance between the subset and the *ideal* task feature. By *Hypothesis III*, no single feature will suffice and a set is required. okay.

Where A is the adjacency matrix of feature distances. That is, the element a_{ij} is the distance from feature i to j. This BIP can be solved in under a second.

The above program finds the minimum covering set for any δ. Since there are only m^2 distances, we can find the minimum δ with binary search, by solving $\mathcal{O}(log(m))$ BIPs. This takes under 5 seconds and the final boolean vector x specifies the feature set of size k that minimizes perceptual risk.

4. Experiments

In this section we describe our experimental setup and present the results from our hypothesis tests and selection module. With 20 vision features and 4 baselines, our approach leads to training between 3-8 seeds per scenario in order to control the false discovery rate [8]. The total number of policies used in the study is about 800 which took 109,639 GPU-hours to train and evaluate.

4.1. Experimental Setup

Environments: We use the Gibson environment [86] which is designed to be *perceptually* similar to the real world. Training in the real world is difficult due to the intrinsic complexity and reproducibility issues, but Gibson reasonably captures the inherent complexity by virtualizing real buildings and is reproducible. Gibson is also integrated with the PyBullet physics engine which uses a fast collision-handling system to simulate dynamics. We perform our study in Gibson but provide a video of the trained policies tested on real robots in the supplementary material.

Train/Test Split: We train and test our agents in two disjoint sets of buildings (Fig. 5). The test buildings are different and completely unseen during training The training space for the visual navigation task covers $40.2m^2$ (square meters) and the testing space covers $415.6m^2$. For local planning and exploration, the train and test spaces cover

[1] For ease of exposition we present a simplified version in the main paper. The full version is similar, but also accounts for feature interactions. See the supplementary material.

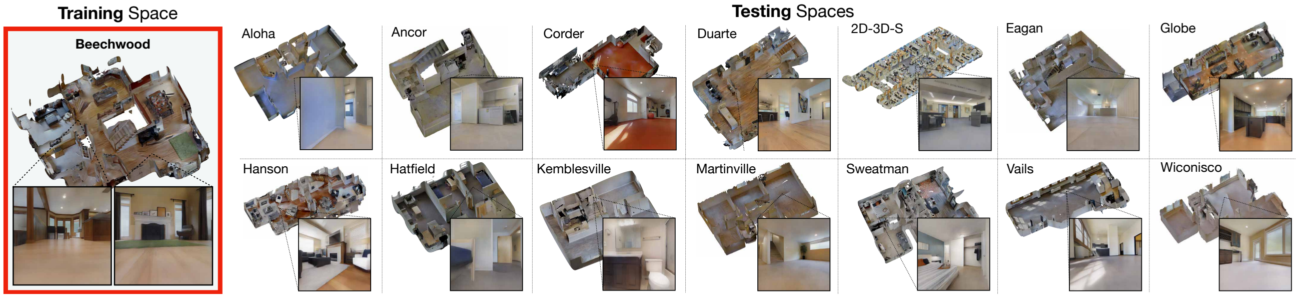

Figure 5. **Visualization of training and test buildings from Gibson database [86].** The training space (on the left, highlighted in red and zoomed) and the testing spaces (remaining on the right). Actual sample observations from agents virtualized in Gibson [86] are shown in the bottom of each box. Results of training and testing in more spaces is provided in the supplementary material.

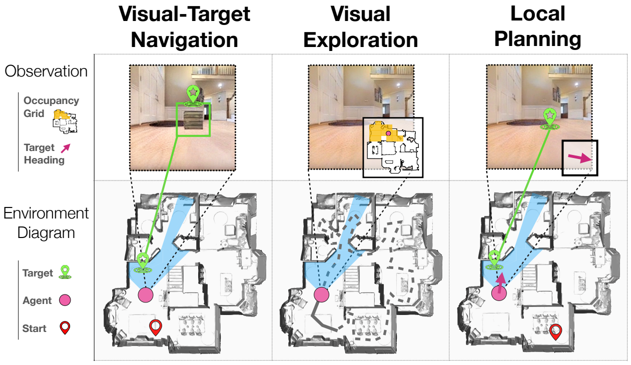

Figure 6. **Active task definitions.** Visual descriptions of the selected active tasks and their implementations in Gibson. Additional observations besides the RGB image are shown in the top row. Note that exploration uses only the revealed occupancy grid and no actual mesh boundaries.

$154.9m^2$ and $1270.1m^2$.

4.1.1 Downstream Active Tasks

In order to test our hypotheses, we sample a few practically useful active tasks: *navigation to a visual target*, *visual exploration*, and *local planning*; depicted in Figure 6 and described below.

Navigation to a Visual Target: In this scenario the agent must locate a specific target object (a wooden crate) as fast as possible with only *sparse rewards*. Upon touching the target there is a large one-time positive reward (+10) and the episode ends. Otherwise there is a small penalty (-0.025) for living. The target looks the same between episodes although the location and orientation of both the agent and target are randomized according to a uniform distribution over a predefined boundary within the floor plan of the space. The agent must learn to identify the target during the course of training. The maximum episode length is 400 timesteps and the shortest path averages around 30 steps.

Visual Exploration: The agent must visit as many **new** parts of the space as quickly as possible. The environment is partitioned into small occupancy cells which the agent "unlocks" by scanning with a myopic laser range scanner. This scanner reveals the area directly in front of the agent for up to 1.5 meters. The reward at each timestep is proportional to the number of newly revealed cells. The episode ends after 1000 timesteps.

Local Planning: The agent must direct itself to a given nonvisual target destination (specified using coordinates) using visual inputs, avoid obstacles and walls as it navigates to the target. This task is useful for the practical skill of local planning, where an agent must traverse sparse waypoints along a desired path. The agent receives dense positive reward proportional to the progress it makes (in Euclidean distance) toward the goal, and is penalized for colliding with walls and objects. There is also a small negative reward for living as in visual navigation. The maximum episode length is 400 timesteps, and the target distance is sampled from a Gaussian distribution, $\mathcal{N}(\mu = 5 \text{ meters}, \sigma^2 = 2 \text{ m})$.

Observation Space: In all tasks, the observation space contains the RGB image and the *minimum* amount of side information needed to feasibly learn the task (Fig. 6). Unlike the common practice, we do not include proprioception information such as the agent's joint positions or velocities or any other side information that could be useful, but is not essential to solving the task. We defer the details of each task's observation space to the supplementary material.

Action Space: We assume a low-level controller for robot actuation, enabling a high-level action space of

$$\mathcal{A} = \{\texttt{turn\_left}, \texttt{turn\_right}, \texttt{move\_forward}\}.$$

Detailed specifications can be found in the supplementary material.

4.1.2 Mid-Level Features

For our experiments, we used representations derived from one of 20 different computer vision tasks (see Fig. 3). This set covers various common modes of computer vision tasks: from texture-based (e.g. denoising), to 3D pixel-level (e.g. depth estimation), to low-dimensional geometry (e.g. room layout), to semantic tasks (e.g. object classification).

We used the networks of [90] trained on a dataset of 4 million static images in of indoor scenes [90]. Each network encoder consists of a ResNet-50 [31] without a global average-pooling layer. This preserves spatial information in the image. The feature networks were all trained using

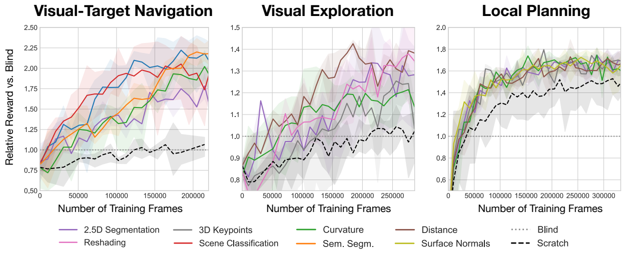

Figure 7. **Sample efficiency of feature-based agents.** Average rewards in the test environment for features and scratch. Feature-based policies learn notably faster.

identical hyperparameters. For a full list of vision tasks, their descriptions, and sample videos of the networks evaluated in our environments, please see the website.

4.1.3 Reinforcement Learning Algorithm

In all experiments we use the common Proximal Policy Optimization (PPO) [67] algorithm with Generalized Advantage Estimation [66]. Due to the computational load of rendering perceptually realistic images in Gibson we are only able to use a single rollout worker and we therefore decorrelate our batches using experience replay and off-policy variant of PPO. The formulation is similar to Actor-Critic with Experience Replay (ACER) [85] in that full trajectories are sampled from the replay buffer and reweighted using the first-order approximation for importance sampling. We include the full formulation, full experimental details, as well as all network architectures in the supplementary material.

For each task and each environment we conduct a hyperparameter search optimized for the *scratch* baseline (see section 4.2). We then fix this setting and reuse it for every feature. This setup favors *scratch* and other baselines that use the same architecture, yet the features outperform them.

4.2. Baselines

We include several control groups as baselines which address possible confounding factors:

Tabula Rasa (Scratch) Learning: The most common approach, *tabula rasa* learning trains the agent from scratch. In this condition (sometimes called *scratch*), the agent receives the raw RGB image as input and uses a randomly initialized AtariNet [51] tower.

Blind Intelligent Actor: The *blind* baseline is the same as *tabula rasa* except that the visual input is a fixed image and does not depend on the state of the environment. A *blind* agent indicates how much performance can be squeezed out of the nonvisual biases, correlations, and overall structure of the environment. For instance, in a narrow straight corridor which leads the agent to the target, there should be a small performance gap between *sighted* and *blind*. The *blind* agent is a particularly informative and crucial baseline.

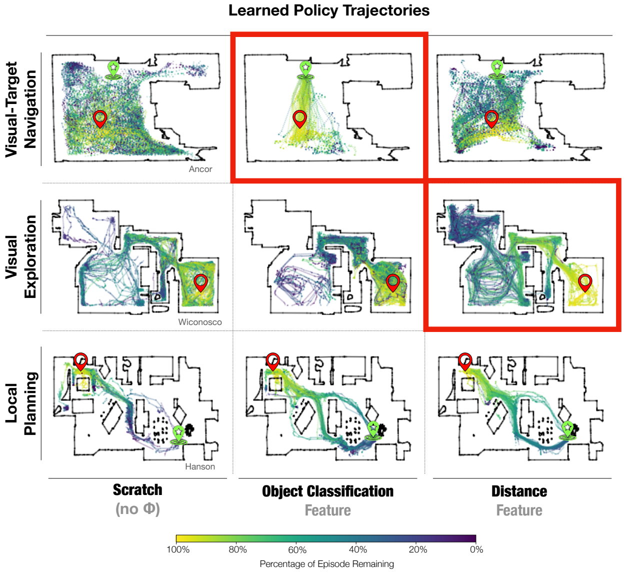

Figure 8. **Agent trajectories in test environment.** Left: The *scratch* policy fails to generalize, inefficiently wandering around the test space. Center: The policy trained with *object classification* recognizes and converges on the navigation target (boxed), but fails to cover the entire space in exploration. Right: *Distance estimation* features only help the agent cover nearly the entire space in exploration (boxed), but fail in navigation unless the agent is nearly on top of the target. Visualizations from all features on all tasks are available on the website.

Random Nonlinear Projections: this is identical to using *mid-level* features, except that the encoder network is randomly initialized and then frozen. The policy then learns on top of this fixed nonlinear projection.

Pixels as Features: this is identical to using *mid-level* features, except that we downsample the input image to the same size as the features (16×16) and use it as the feature. This addresses whether the feature readout network could be an improvement over AtariNet which is used for scratch for tractability.

Random Actions: uniformly randomly samples from the action space. If random actions perform well then the there is not much to be gained from learning.

State-of-the-Art Feature Learning: offers a comparison of mid-level visual features against several other (not necessarily vision-centric) approaches. We compare against several state-of-the-art feature methods, including dynamic modeling [53, 68, 36], curiosity [58], DARLA [33], and ImageNet pretraining [42], enumerated in Figure 10.

4.2.1 Quantification

RL results are typically communicated in terms of absolute reward. However, absolute reward values are uncalibrated and a high value for one task is not necessarily impressive in another. One way to calibrate rewards according to task difficulty is by comparing to a control that cannot access the state of the environment. Therefore, we propose the *reward relative to blind*:

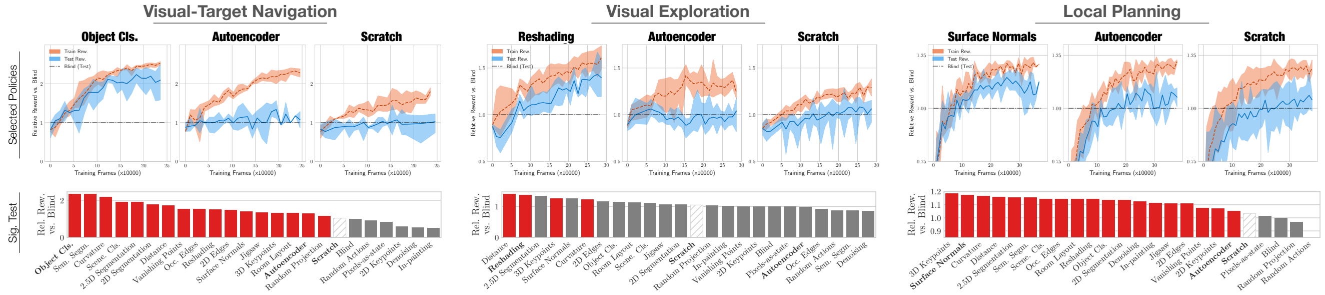

Figure 9. **Mid-level feature generalization.** The curve plots above show training and test performance of *scratch* vs. best features throughout training. For all tasks there is a significant gap between train/test performance for *scratch*, and a much smaller one for the best feature. Note that feature-based agents generalize well while *scratch* does not. **This underscors the importance of separating the train and test environment in RL**. The bar charts show agent performance in the test environment. Agents significantly better than *scratch* are shown in red.[2]

$$RR_{\text{blind}} = \frac{r_{\text{treatment}} - r_{\min}}{r_{\text{blind}} - r_{\min}} \tag{1}$$

as a calibrated quantification. A *blind* agent always achieves a relative reward of 1, while a score > 1 indicates a relative improvement and score < 1 indicates this agent performs worse than a *blind* agent. We find this quantification particularly meaningful since we found agents trained from scratch often memorize the training environment, performing no better than *blind* in the test setting (see Fig. 9). We provide the raw reward curves in the supplementary material for completeness.

4.3. Experimental results on hypothesis testing I-III

In this section we report our findings on the effect of mid-level representations on sample efficiency and generalization. All results are evaluated in the *test* environment with multiple random seeds, unless otherwise explicitly stated. When we use a significance test we opt for a nonparametric approach, sacrificing statistical power to eliminate assumptions on the relevant distributions[2].

4.3.1 Hypothesis I: Sample Complexity Results

We find that for each of our active tasks, several feature-based agents learn significantly faster than *scratch*. We evaluated twenty different features against the four control groups on each of our tasks: *visual-target navigation*, *visual exploration*, and *local planning*. Evaluation curves for the five top-performing features appear in Figure 7. Randomly sampled trajectories in Figure 8 highlight how agents trained using features have qualitatively different performance than agents trained *tabula rasa*.

4.3.2 Hypothesis II: Generalization Results

We find that for each of our tasks, several feature-based agent achieved higher final performance than policies trained *tabula rasa*. We explore conditions when this may *not* hold in Section 4.5.

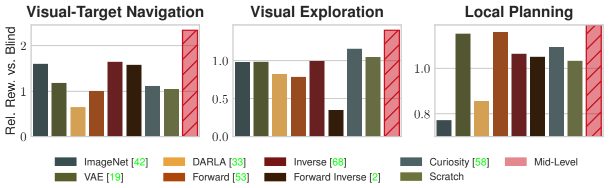

Figure 10. **Comparison between our mid-level vision and state-of-the-art feature learning methods**. The vertical axis shows the achieved reward. Note the large gap between mid-level features and the alternatives.

Large-Scale Analysis: On each task, some features outperform *tabula rasa* learning. Figure 9 shows features that outperform *scratch* and those that do so with high confidence are highlighted in red. Significance tests[2] reveal that the probability of so many results being due to noise is < 0.002 per task ($< 10^{-6}$ after the analysis in Sec. 4.3.3).

Mind the Gap: All agents exhibited some gap between training and test performance, but agents trained from scratch seem to overfit completely—rarely doing better than blind agents in the test environment. The plots in Figure 9 show representative examples of this disparity over the course of training. Similarly, some common features like *Autoencoders* and *VAEs* have strong training curves that belie exceptionally weak test-time performance.

4.3.3 Hypothesis III: Rank Reversal Results

We found that there may not be one or two single features that consistently outperform all others. Instead, the choice of pretrained features should depend upon the downstream task. This experiment demonstrates this dependence by exhibiting a case of *rank reversal*.

Case Study: The top-performing exploration agent used *Distance Estimation* features, perhaps because an effective explorer needs to identify doorways to new, open spaces. In contrast, the top navigation agent used *Object Classification* features—ostensibly because the agent needs to identify the target crate. Despite being top of their class on their pre-

[2] We use the nonparametric Mann-Whitney U test, correcting for multiple comparisons by controlling the False Discovery Rate ($Q = 20\%$) with Benjamini-Hochberg [8].

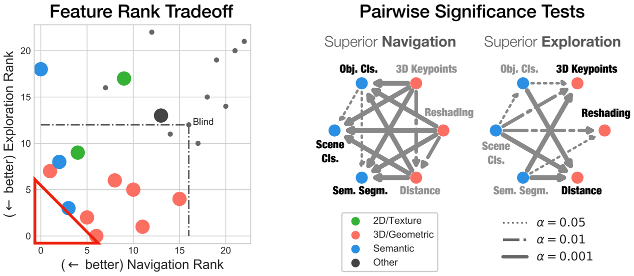

Figure 11. **Rank reversal in visual tasks and absence of universal features.** Right: Scatter plot showing feature ranks in navigation (x-axis) and exploration (y-axis). The fact that there is no feature on the bottom left corner (marked with the red triangle) indicates there was no universal feature. In fact, there is no *almost* universal feature, and maximizing F-score requires giving up 3-4 ranks for each task. Left: 60 pairwise significance tests between the three best features on each task quantify this result. Arrows represent a significant result and they point towards the feature that performed better on the downstream task. **Heavier arrows** denote higher significance (lower α-level). Lack of an arrow indicates that performances were statistically indistinguishable. The essentially complete bipartite structure in the graphs shows that navigation is characteristically semantic while exploration is geometric.

ferred tasks, neither feature performed particularly well on the other task. This result was statistically significant (in both directions) at the $\alpha = 0.0005$ level. Fig. 8 visualizes these difference by plotting randomly sampled agent trajectories in a test environment.

Ubiquity of Rank Reversal: The trend of rank reversal appears to be a widespread phenomenon. Fig. 11 shows the results from sixty pairwise significance tests, revealing that semantic features are useful for navigation while geometric features are useful for exploration, and that the semantic/geometric distinction is highly predictive of final performance. Figure 10 shows that state-of-the-art representation learning methods ares similarly task-specific, but the *best* feature outperforms them by a large margin.

4.4. Max-Coverage Feature Set Analysis

The solver described in Section 3.3 outputs a set that unifies several useful mid-level vision tasks without sacrificing generality. The experimental results in terms of achieved reward by each feature set (with size $k = 1$ to 4) is reported in Figure 12.

Performance: With only k =4 features, our Max-Coverage Feature Set is able to nearly match or exceed the performance of the *best* task-specific feature—even though this set is agnostic to our choice of active tasks.

Sample Efficiency: Due to a larger input space and a larger architecture, we expected worse sample efficiency compared to the best single-feature policy. However, we did not find a noticeable difference.

Practicality: This module is able to combine structured sources of information into a compressed representation for use with RL. By simply replacing the raw pixel observation with our max-coverage feature set, practitioners can gain

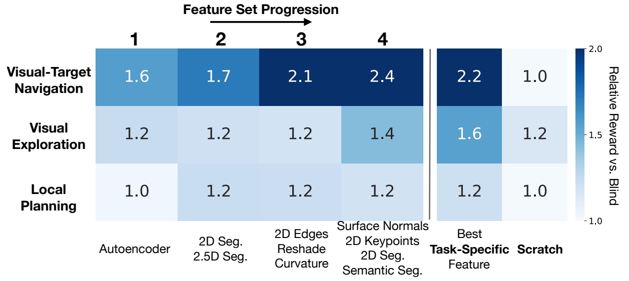

Figure 12. **Evaluation of max-coverage feature sets.** The reward (relative to *blind*) is shown in each box. The left four columns show the performance of the agents trained with the max-coverage feature set, as the set size increases from k=1 to 4. The two right columns are baselines. The baselines are trained for longer so that all policies in this figure saw the same amount of data.

the benefits of mid-level vision. The structure also lends itself well to model-parallelism on modern computational architectures.

4.5. Universality Experiments

4.5.1 Universality in Additional Buildings

We repeated our testing in 9 other buildings to account for the possibility that our main test building is anomalous in some way. We found that the reward in our main test building and in the 9 other buildings was extremely strongly correlated with a Spearman's ρ of 0.93 for navigation and 0.85 for exploration. Full experimental setup and results are included in the supplementary material.

4.5.2 Universality in Additional Simulators

To evaluate whether our findings are an artifact of the Gibson environment, we tested in an additional environment by implementing navigation and exploration in a second 3D simulator, VizDoom [39]. We found that features which perform well in Gibson also tend to perform well in VizDoom. We also replicated our rank reversal findings (with high confidence), including the geometric/semantic distinction for exploration/navigation and the lack of a *universal feature*. Here, too, maximizing the combined score requires choosing the third- or fourth-best feature for any given task.

In addition, only feature-based agents were able to generalize without texture randomization during training. Once we added in randomized training textures that resembled the test textures (in effect, making $\mathcal{P}$ and $\mathcal{Q}$ more similar), this distinction disappeared. Our findings do not contradict the general usefulness of RL in the limit of infinite and varied data. Rather, they indicate that to use RL in the (real) world of limited data, we need to introduce learning biases that put their thumb on the scale. The complete VizDoom universality experiments, as well as relevant plots, detailed descriptions of task implementations, and train/test splits are in the supplementary material.

5. Conclusion and Limitations

This paper presented an approach for using visual biases to learn robotic policies and demonstrated its utility in providing learning biases that improve generalization and reduce sample complexity. We showed that the correct choice of feature depends on the downstream task, and used this fact to refine and generalize our approach: introducing a principled method for selecting a general-purpose feature set. The solver-selected feature sets outperformed state-of-the-art pretraining methods and used at least an order of magnitude less data than learning from scratch—while simultaneously achieving higher final performance.

A great deal of additional research is possible along this direction. The relationship between visual biases and active tasks is itself an interesting object of study, and a better understanding of these dynamics could lead to more general visual abstractions, as well as ones that are explicitly adapt to specific downstream tasks. In this work, we made a number of simplifying assumptions that are worth noting:

Locomotive Tasks: Our selection of active tasks was primarily oriented around locomotion. Though locomotion is a significant enough problem, our study does not necessarily convey conclusions about mid-level vision's utility on other important active tasks, such as manipulation.

Model Dependence: We adopted neural networks as our function class. Though we validated the stability of our findings on additional environments and against several tasks, in principle our findings could be different if we used another model such as nearest neighbors.

Reinforcement Learning: Given that we used RL as our experimental platform, our findings are enveloped by the limitations of existing RL methods, e.g. difficulties in long-range exploration or credit assignment with sparse rewards.

Lack of Guarantees: Our approach is primarily empirical. Our measures of agent success and perceptual distance were both derived from experimental results. Successfully predicting agent performance in a test setting would be important for safely deploying robots in a new environment.

Limited Representation Set: We used a fixed set of mid-level features, and the best-performing feature necessarily depends on the choice of this set. In addition, we froze the feature weights, limiting how expressive a feature-based policy could possibly be. Relaxing this constraint could improve the worst-case performance to be similar to that of *tabula rasa* learning.

Lifelong Learning: Our mid-level feature set is fixed. How to continually update the visual estimators and how to incrementally expand the dictionary are important future research questions.

Acknowledgements We gratefully acknowledge the support of ONR MURI (N00014-14-1-0671), NVIDIA NGC beta, NSF (IIS-1763268), and TRI. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM. 2

[2] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *CoRR*, abs/1606.07419, 2016. 2, 7

[3] S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 39.1–39.26, Edinburgh, Scotland, 25–27 Jun 2012. PMLR. 2

[4] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. 2

[5] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, Mar. 2003. 2

[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010. 2

[7] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[8] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. 4, 7

[9] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. 2

[10] T. Chen, S. Gupta, and A. Gupta. Learning Exploration Policies for Navigation. *arXiv e-prints*, page arXiv:1903.01959, Mar 2019. 2

[11] F. Codevilla, A. López, V. Koltun, and A. Dosovitskiy. On offline evaluation of vision-based driving models. *CoRR*, abs/1809.04843, 2018. 2

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2

[13] C. Devin, P. Abbeel, T. Darrell, and S. Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118, May 2018. 2

[14] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2

[15] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016. 2

[16] A. Dosovitskiy and V. Koltun. Learning to act by predicting the future. *CoRR*, abs/1611.01779, 2016. 2

[17] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016. 2

[18] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. 2

[19] S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 7

[20] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. 2

[21] C. Finn, K. Xu, and S. Levine. Probabilistic Model-Agnostic Meta-Learning. *ArXiv e-prints*, June 2018. 2

[22] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *CoRR*, abs/1709.04905, 2017. 2

[23] J. Fu, J. D. Co-Reyes, and S. Levine. EX2: exploration with exemplar models for deep reinforcement learning. *CoRR*, abs/1703.01260, 2017. 2

[24] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, Jan 1992. 1

[25] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 2

[26] A. Giusti, J. Guzzi, D. C. Cirean, F. He, J. P. Rodrguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, July 2016. 2

[27] E. Grant, C. Finn, S. Levine, T. Darrell, and T. L. Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *CoRR*, abs/1801.08930, 2018. 2

[28] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. *CoRR*, abs/1702.03920, 2017. 2

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. 2

[30] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 2

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 5

[32] 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*, pages 770–778, 2016. 2

[33] I. Higgins, A. Pal, A. A. Rusu, L. Matthey, C. P. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. *arXiv e-prints*, page arXiv:1707.08475, Jul 2017. 6, 7

[34] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2

[35] R. Hu, P. Dollár, K. He, T. Darrell, and R. B. Girshick. Learning to segment every thing. *CoRR*, abs/1711.10370, 2017. 2

[36] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992. 2, 6

[37] A. Kanazawa, J. Zhang, P. Felsen, and J. Malik. Learning 3d human dynamics from video. *CoRR*, abs/1812.01601, 2018. 2

[38] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine. Generalization through Simulation: Integrating Simulated and Real Data into Deep Reinforcement Learning for Vision-Based Autonomous Flight. *arXiv e-prints*, page arXiv:1902.03701, Feb 2019. 2

[39] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski. Vizdoom: A doom-based AI research platform for visual reinforcement learning. *CoRR*, abs/1605.02097, 2016. 8

[40] T. Kim, J. Yoon, O. Dia, S. Kim, Y. Bengio, and S. Ahn. Bayesian Model-Agnostic Meta-Learning. *ArXiv e-prints*, June 2018. 2

[41] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2, 6, 7

[43] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 2

[44] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–101, 2016. 2

[45] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015. 1, 2

[46] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015. 2

[47] M. Long and J. Wang. Learning transferable features with deep adaptation networks. *CoRR*, abs/1502.02791, 2015. 2

[48] L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR 2017*, 2017. 2

[49] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. Meta-learning with temporal convolutions. *CoRR*, abs/1707.03141, 2017. 2

[50] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. 2

[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015. 1, 2, 6

[52] A. Mousavian, A. Toshev, M. Fiser, J. Kosecka, and J. Davidson. Visual representations for semantic target driven navigation. *CoRR*, abs/1805.06066, 2018. 2

[53] J. Munk, J. Kober, and R. Babuka. Learning state representation for deep actor-critic control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4667–4673, Dec 2016. 6, 7

[54] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 2

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

[56] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. *arXiv preprint arXiv:1708.06734*, 2017. 2

[57] A. Odena. Semi-Supervised Learning with Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1606.01583, Jun 2016. 2

[58] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017. 2, 6, 7

[59] J. W. Peirce. Understanding mid-level representations in visual processing. *Journal of Vision*, 15(7):5–5, 06 2015. 2

[60] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. *CoRR*, abs/1812.01754, 2018. 2

[61] A. Raffin, A. Hill, K. R. Traoré, T. Lesort, N. D. Rodríguez, and D. Filliat. Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. *CoRR*, abs/1901.08651, 2019. 2

[62] A. Raffin, A. Hill, R. Traoré, T. Lesort, N. D. Rodríguez, and D. Filliat. S-RL toolbox: Environments, datasets and evaluation metrics for state representation learning. *CoRR*, abs/1809.09369, 2018. 2

[63] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. *CoRR*, abs/1707.02920, 2017. 2

[64] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 2

[65] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *CoRR*, abs/1712.02560, 2017. 2

[66] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015. 6

[67] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. 2, 6

[68] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *CoRR*, abs/1612.07307, 2016. 6, 7

[69] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer-Verlag, Berlin, Heidelberg, 2007. 2

[70] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. *Indoor Segmentation and Support Inference from RGBD Images*, pages 746–760. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2

[71] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. 2

[72] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 73–86, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[73] E. S. Spelke and K. D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007. 2

[74] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4732–4741, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[75] B. Sun and K. Saenko. Deep CORAL: correlation alignment for deep domain adaptation. *CoRR*, abs/1607.01719, 2016. 2

[76] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. 2

[77] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. 2

[78] A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008. 3

[79] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. *CoRR*, abs/1510.02192, 2015. 2

[80] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *CoRR*, abs/1702.05464, 2017. 2

[81] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 2

[82] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 2

[83] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM. 2

[84] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[85] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas. Sample efficient actor-critic with experience replay. *CoRR*, abs/1611.01224, 2016. 6

[86] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 2, 3, 4, 5

[87] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017. 2

[88] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi. Visual semantic navigation using scene priors. *CoRR*, abs/1810.06543, 2018. 2

[89] T. Yu, P. Abbeel, S. Levine, and C. Finn. One-shot hierarchical imitation learning of compound visuomotor tasks. *CoRR*, abs/1810.11043, 2018. 2

[90] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 2, 3, 4, 5

[91] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. 2

[92] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696, Sept 2009. 2

[93] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[94] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi. Visual semantic planning using deep successor representations. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2