The (Un)Surprising Effectiveness of Pre-Trained Vision Models for Control

Anonymous Authors[1]

Abstract

Recent years have seen the emergence of pre-trained representations as a powerful abstraction for AI applications in computer vision, natural language, and speech. However, policy learning for control is still dominated by a tabula-rasa learning paradigm, with visuo-motor policies often trained from scratch using data from deployment environments. In this context, we revisit and study the role of pre-trained visual representations for control, and in particular representations trained on large-scale computer vision datasets. Through extensive empirical evaluation in diverse control domains (Habitat, DeepMind Control, Adroit, Franka Kitchen), we isolate and study the importance of different representation training methods, data augmentations, and feature hierarchies. Overall, we find that pre-trained visual representations can be competitive or even better than ground-truth state representations to train control policies. This is in spite of using only out-of-domain data from standard vision datasets, without any in-domain data from the deployment environments.

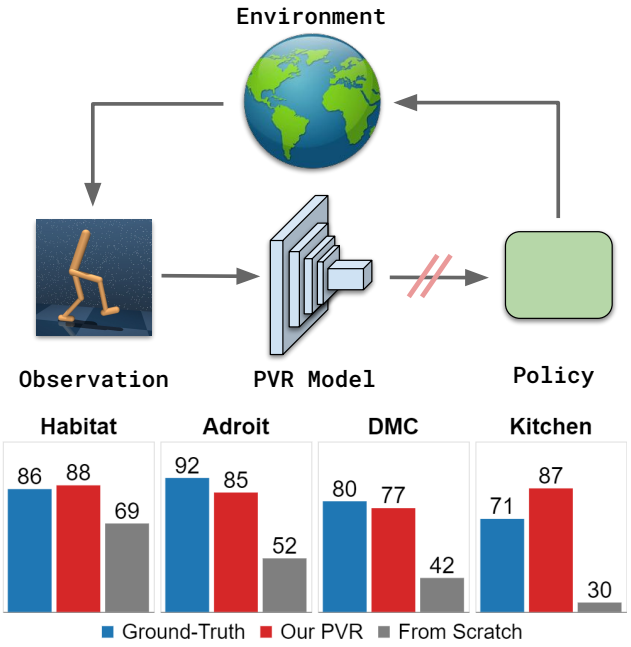

Figure 1: **(Top)** In our paradigm, a pre-trained vision model is used as a perception module for the policy. The model is frozen and not further trained during policy updates. Its output, namely the pre-trained visual representation (PVR), serves as state representation and policy input. **(Bottom)** Our PVR is competitive with ground-truth features for training policies with imitation learning, in spite of being pre-trained on out-of-domain data. By contrast, the classic approach of training an end-to-end visuo-motor policy from scratch fails with the same amount of imitation data.

1. Introduction

Representation learning has emerged as a key component in the success of deep learning for computer vision, natural language processing (NLP), and speech processing. Representations trained using massive amounts of labeled (Krizhevsky et al., 2012; Sun et al., 2017; Brown et al., 2020) or unlabeled (Devlin et al., 2019; Goyal et al., 2021) data have been used "off-the-shelf" for many downstream applications, resulting in a simple, effective, and data-efficient paradigm. By contrast, policy learning for control is still dominated by a "tabula-rasa" paradigm where an agent performs millions or even billions of interactions with an environment to learn task-specific visuo-motor policies from scratch (Espeholt et al., 2018; Wijmans et al., 2020; Yarats et al., 2021b).

In this paper, we take a step back and ask the following fundamental question. Why have pre-trained visual representations, like those trained on ImageNet, not found widespread success in control despite their ubiquitous usage in computer vision? Is it because control tasks are too different from vision tasks? Or because of the domain gap in the visual characteristics? Or is it that "the devil lies in the details", and we are failing to consider some key components? We note that dataset domain gap is not a core issue in computer

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

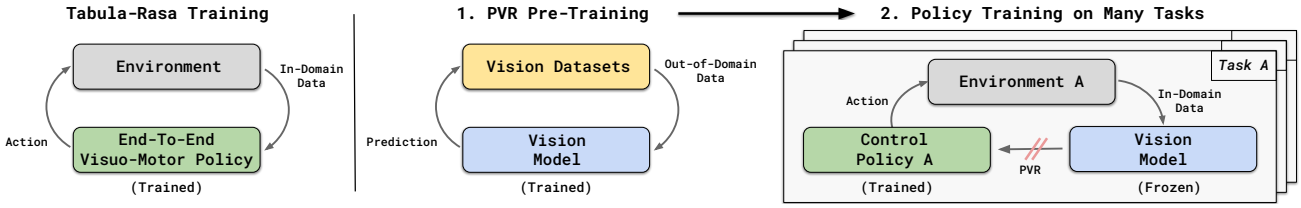

Figure 2: **Classic training paradigm (left) vs. ours (right)**. In tabula-rasa training, the perception module is part of the control policy and is trained from scratch on data from the environment. By contrast, in our paradigm the perception module is detached from the policy. First, it is trained once on out-of-domain data (e.g., ImageNet) and frozen. Then, given some tasks, control policies are trained on the deployment environments re-using the same frozen perception module.

vision. For instance, ImageNet-trained models have been shown to transfer to a variety of different tasks like human pose estimation (Cao et al., 2017). In this context, we aim to investigate the following fundamental question.

Can we make a single vision model, pre-trained entirely on out-of-domain datasets, work for different control tasks?

To answer this question, we consider a large collection of pre-trained visual representation (PVR) models commonly used in computer vision, and investigate how such models can be used as frozen perception modules for control tasks, as depicted in Figure 2. We perform a series of experiments to understand the effectiveness of these representations in four well-known domains that require visuo-motor control policies: Habitat (Savva et al., 2019), DeepMind Control (Tassa et al., 2018), Adroit dexterous manipulation (Rajeswaran et al., 2018), and Franka kitchen (Gupta et al., 2019). Our investigation reveals very surprising results[1] that can be summarized as follows.

- Our main finding is that frozen PVRs trained on completely out-of-domain datasets can be competitive with or even outperform ground-truth state features for training policies (with imitation learning). We emphasize that these vision models have never seen even a single frame from our evaluation environments during pre-training.
- Self-supervised learning (SSL) provides better features for control policies compared to supervised learning.
- Crop augmentations appear to be more important in SSL for control compared to color augmentations. This is consistent with a number of "on-the-fly" representation learning works that primarily employ crop augmentations (Srinivas et al., 2020; Yarats et al., 2021b).
- Early convolution layer features are better for fine-grained control tasks (MuJoCo) while later convolution layer features are better for semantic tasks (Habitat ImageNav).
- By combining features from multiple layers of a pre-trained vision model, we propose a single PVR that is competitive with or outperform ground-truth state features in **all** the domains we study.

[1]We argue that our findings are surprising in the context of representation learning for control. At the same time, the success of PVRs should have been unsurprising considering their widespread success and use in computer vision.

2. Related Work

Representation Learning. Pre-training representations and transfering them to downstream applications is an old and vibrant area of research in AI (Hinton & Salakhutdinov, 2006; Krizhevsky et al., 2012). This approach gained renewed interest in the fields of computer vision, speech, and NLP with the observation that representations learned by deep networks transfer remarkably well to downstream tasks (Girshick et al., 2014; Devlin et al., 2019; Baevski et al., 2020), resulting in improved data efficiency and/or improved performance (Goyal et al., 2019).

Focusing on computer vision, representations can be learned either through supervised methods, such as ImageNet classification (Krizhevsky et al., 2012; Russakovsky et al., 2015), or through self-supervised methods that do not require any labels (Doersch et al., 2015; Chen et al., 2020; Purushwalkam & Gupta, 2020). The learned representations can be used "off-the-shelf", with the representation network frozen and not adapted to downstream tasks. This approach has been successfully used in object detection (Girshick et al., 2014; Girshick, 2015), segmentation (He et al., 2017), captioning (Vinyals et al., 2016), and action recognition (Hara et al., 2018). In this work, we investigate if frozen pre-trained visual representations can also be used for policy learning in control tasks.

Policy Learning. Reinforcement learning (RL) (Sutton & Barto, 1998) and imitation learning (IL) (Abbeel & Ng, 2004) are two popular classes of approaches for policy learning. In conjunction with neural network policies, they have demonstrated impressive results in a wide variety of control tasks spanning locomotion, whole arm manipulation, dexterous hand manipulation, and indoor navigation (Heess et al., 2017; Rajeswaran et al., 2018; Peng et al., 2018; Wijmans et al., 2020; OpenAI et al., 2020; Weihs et al., 2021).

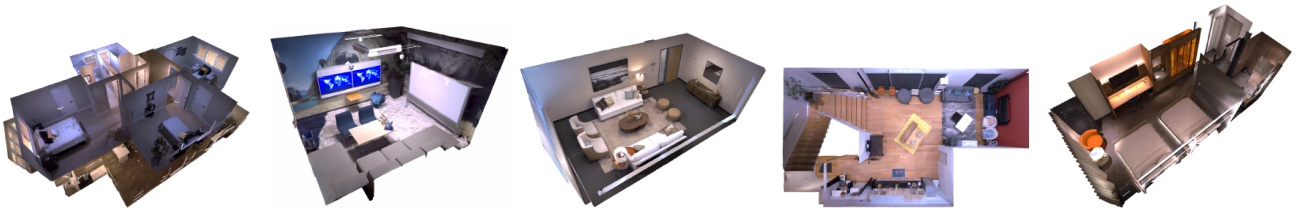

Figure 3: **Real-world scenes from the Replica dataset used in Habitat.** The agent has to reach target locations from anywhere on the scene. Its perception is based on its egocentric view of the scene and an image showing the target location. Only ground-truth state features explicitly inform the agent about its position, the target coordinates, and the scene it is in.

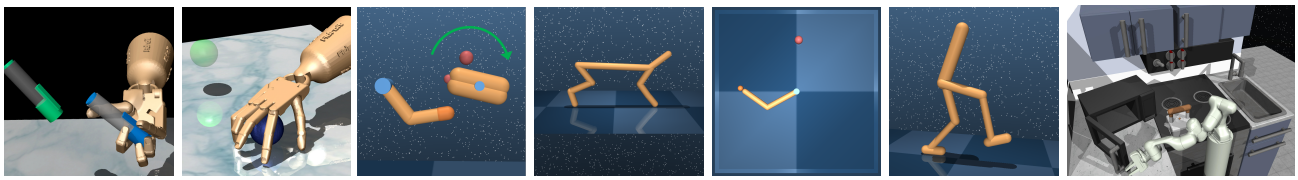

Figure 4: **MuJoCo tasks span three domains.** In Adroit (left), the agent has to learn dexterous hand manipulation behaviors like grasping and in-hand manipulation. In the DeepMind Control suite (center), it needs to learn low-level locomotion and manipulation behaviors. In Franka Kitchen (right), it has to reconfigure objects in a kitchen using a Franka arm.

In this work, we focus on learning visuo-motor policies using IL. A large body of work in IL and RL for continuous control has focused primarily on learning from ground-truth state features (Schulman et al., 2015; Lillicrap et al., 2016; Ho & Ermon, 2016). While such privileged state information may be available in simulation or motion capture systems, it is seldom available in real-world settings. This has motivated researchers to investigate continuous control from visual inputs by building upon ideas like data augmentations (Laskin et al., 2020; Yarats et al., 2021b), contrastive learning (Srinivas et al., 2020; Zhang et al., 2021), or predictive world models (Hafner et al., 2020; Rafailov et al., 2021). However, these works still learn representations from scratch using frames from the deployment environments.

Pre-trained Visual Encoders in Control. The use of pre-trained vision models in control tasks has received limited attention. Stooke et al. (2021) pre-trained representations in DeepMind Control suite, and evaluated downstream policy learning in the same domain. By contrast, we study the use of representations learned using out-of-domain datasets, which is a more scalable paradigm that is not limited by frames from the deployment environment. Khandelwal et al. (2021) studied the use of pre-trained CLIP embeddings for visual navigation tasks and reported improved results over encoders trained from scratch. Similarly, Shah & Kumar (2021) studied ImageNet pre-trained ResNet representations, and found promising results in Adroit but negative results in DeepMind control suite. Compared to these works, our study is more exhaustive: it spans four visually diverse domains, a larger collection of pre-trained representations, and different forms of visual invariances stemming from augmentations and layers. Ultimately, we find that a single pre-trained representation can be successful for all the domains we study despite their visual and task-level diversity.

3. Experiments Setup

3.1. Environments

Habitat (Savva et al., 2019) is a home assistant robotics simulator showcasing the generality of our paradigm to a visually realistic domain. The agent is trained to navigate the five Replica scenes (Straub et al., 2019) shown in Figure 3. We consider the `ImageNav` task, where the agent is given two images at each timestep corresponding to the agent's current view and the target location.

DeepMind Control (DMC) Suite (Tassa et al., 2018) is a collection of environments simulated in MuJoCo (Todorov et al., 2012), and a widely studied benchmark in continuous control. In our evaluation, we consider five tasks from the suite: `Finger-Spin`, `Reacher-Hard`, `Cheetah-Run`, `Walker-Stand`, and `Walker-Walk`. These tasks are illustrated in Figure 4 and require the agent to learn low-level locomotion and manipulation skills.

Adroit (Rajeswaran et al., 2018) is a suite of tasks where the agent must control a 28-DoF anthropomorphic hand to perform a variety of dexterous tasks. We study the two hardest tasks from this suite: `Relocate` and `Reorient Pen`, depicted in Figure 4. The policy is required to perform

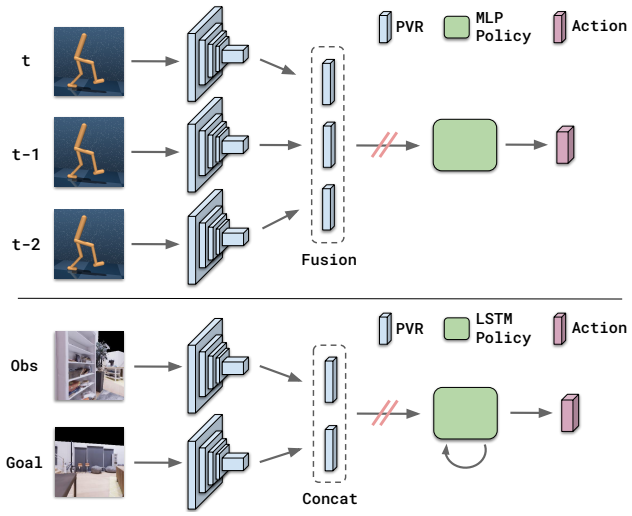

Figure 5: **Learning architecture for MuJoCo (top) and Habitat (bottom)**. In the former, the last three frames are fed to the vision model to obtain PVR embeddings. These are then fused (Shang et al., 2021) and passed to the control policy. In the latter, we embed two images –the agent's current view of the scene and the a view of the target location. The PVR embeddings are concatenated and passed to the control policy. See Appendix A for more details.

goal-conditioned behaviors where the goals (e.g., desired location/orientation for the object) has to be inferred from the scene. These environments are also simulated in MuJoCo, and are known to be particularly challenging.

Franka Kitchen (Gupta et al., 2019) requires to control a simulated Franka arm to perform various tasks in a kitchen scene. In this domain, we consider five tasks: `Microwave`, `Left-Door`, `Right-Door`, `Sliding-Door`, and `Knob-On`. Consistent with use in other benchmarks like D4RL (Fu et al., 2020), we randomize the pose of the arm at the start of each episode, but not the scene itself.

3.2. Models

We investigate the efficacy of representations learned using a variety of methods including approaches that rely on supervised learning (SL) and self-supervised learning (SSL).

ResNet (RN) (He et al., 2016) refers to residual networks, a class of models widely used in computer vision. Typically, these networks are pre-trained using SL on ImageNet (Deng et al., 2009), and can have different size. In our experiments, we use ResNet-50 (RN50) and ResNet-34 (RN34).

Momentum Contrast (MoCo) (He et al., 2020) is a recently proposed SSL method relying on the instance discrimination task to learn representations. These representations have shown competitive performance on numerous downstream tasks in computer vision like image classification, object detection, and instance segmentation. MoCo uses multiple artificial augmentations like cropping, horizontal flipping, and color jitter in order to synthesize multiple views for a single image. The combination of all these augmentations is referred to as 'Aug+'.

Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) jointly trains a visual and textual representation using a collection of image-text pairs from the web. The learned representation has demonstrated impressive semantic discriminative power, zero-shot learning capabilities, and generalization across numerous domains of visual data.

Random Features. As baseline, we consider a randomly initialized convolutional neural network of five layers (each with 32 filters, 3×3 kernel, stride 2, and padding 1) with ELU activation at each layer. Similarly to previous models, this network is frozen and not updated during learning.

From Scratch. We also compare with the classic end-to-end policy learning approach, where the perception module is also trained as part of the policy. We argue that this is an inefficient approach to learning visuo-motor policies, as learning good visual encoders is known to be data-hungry.

Ground-Truth Features. Simulators can provide compact ground-truth features describing the full state of the agent and environment. Such features are hard to estimate in real-world tasks, especially in unstructured environments. Thus, we can view these features as an "oracle" baseline that we strive to compete with.

3.3. Policy Learning and Evaluation with PVRs

The aforementioned models are used as frozen perception modules for the control policy. The policy is trained by imitating optimal trajectories, and the performance is estimated using evaluation rollouts in the environments.

- In Habitat, training trajectories are generated using its native solver that returns the shortest path between two locations. We collect 10,000 trajectories per scene, for a total of ∼2.1 million data points. A policy is successful if the agent reaches the destination within the steps limit.
- In MuJoCo, training trajectories are collected using a state-based optimal policy trained using RL. We collect different trajectories for the domains based on our estimate of task difficulty and horizon. In the case of Adroit and Kitchen, we report policy success percentage provided by the environments. For DMC, we report the policy return rescaled to be in the range of $[0, 100]$.

The learning setup is summarized in Figure 5. In line with standard design choices, we use an LSTM policy to incorporate trajectory history in case of Habitat (Raileanu & Rocktäschel, 2020; Parisi et al., 2021) and use an MLP with fixed history window in case of MuJoCo tasks (Yarats et al., 2021b; Laskin et al., 2020).

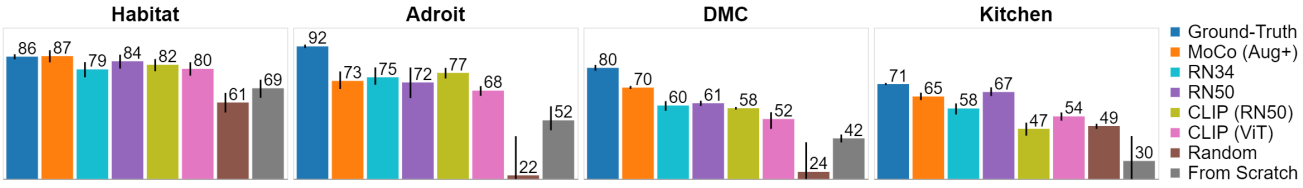

Figure 6: **Success rate of off-the-shelf PVRs**. Numbers at the top of the bar report mean values over five seeds, while thin black lines denote 95% confidence intervals. Any PVR is better than training the perception and control network end-to-end from scratch. In Habitat, MoCo matches the performance of ground-truth features. In MuJoCo, these PVRs are unable to match the ground-truth features off the shelf.

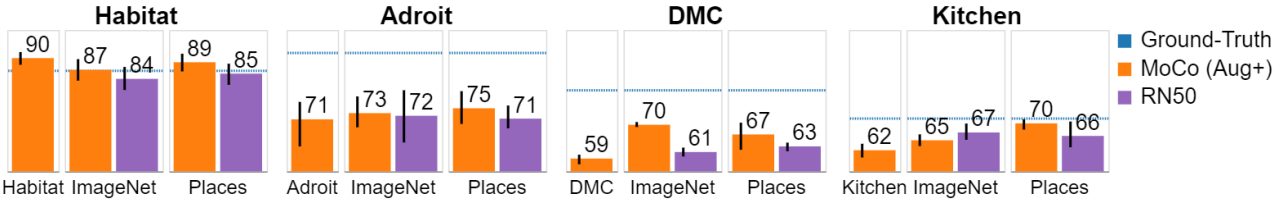

Figure 7: **In-domain vs. out-of-domain training datasets.** Training PVRs on in-domain data does not help achieving better performance. In MuJoCo it even worsen it. If not the domain gap, what is the primary reason of PVRs failures?

4. Experiments Results and Discussion

In the previous sections, we explained the experimental setup for training control policies using behavior cloning, and the testing environments from Habitat and MuJoCo. In this section, we experimentally study the performance of PVRs outlined in Section 3. In particular, we study how well these representations perform out of the box, and how we could potentially improve or customize them, with the ultimate goal of better understanding the relationship between visual perception and control policies. For hyper-parameter details see Appendix A.

4.1. How do Off-the-shelf Vision Models Perform for Control Tasks?

We first study how the pre-trained vision models presented in Section 3.2 perform off-the-shelf for our control task suite. That is, we download these models –pre-trained on ImageNet (Deng et al., 2009)– and pass their output as representations to the control policy. The results are summarized in Figure 6. Firstly, we find that **any** PVR is clearly better than both frozen random features and learning the perception module from scratch, in the small-dataset regime we study. This is perhaps not too surprising, considering that representation learning is known to be data intensive.

However, Figure 6 also provides mixed results as no PVR is clearly superior to any other across all four domains. Nonetheless, on average, SSL models (MoCo) are better than SL models (RN50, CLIP). In particular, MoCo is competitive with ground-truth features in Habitat, but no off-the-shelf PVR can match the ground-truth features in MuJoCo. Why is this so, and can we customize the PVRs to perform better for all control tasks? We investigate different hypotheses and customizations in the following sub-sections.

4.2. Datasets and Domain Gap

The PVRs evaluated above were the representations of vision models trained on ImageNet (Deng et al., 2009). Clearly, the visual characteristics of ImageNet is very different from Habitat and MuJoCo domains. Could this domain gap be the reason why PVRs are not competitive with ground-truth features in all domains? To investigate this, we introduce new datasets for pre-training the vision models. The first is Places (Zhou et al., 2017), another out-of-domain dataset like ImageNet that is widely used in computer vision. While ImageNet is more object-centric, Places is more scene-centric, as it was developed for scene recognition. The other datasets are in-domain frames from Habitat and MuJoCo, i.e., they each contain only images from the deployment environment.

For the Places dataset, we pre-train both supervised and self-supervised vision models. For the Habitat and MuJoCo datasets, we only pre-train self-supervised models since no direct supervision is available. Furthermore, pre-training models using environment data (Habitat, MuJoCo) requires design decisions like data collection policy and dataset size. For sake of simplicity, we collect trajectories using the same expert policies used for IL. Larger or more diverse datasets from these environments may further improve the quality of the pre-trained representations, but run contrary to the motivation of simple and data-efficient learning for control.

Figure 7 summarizes the results for the aforementioned rep-

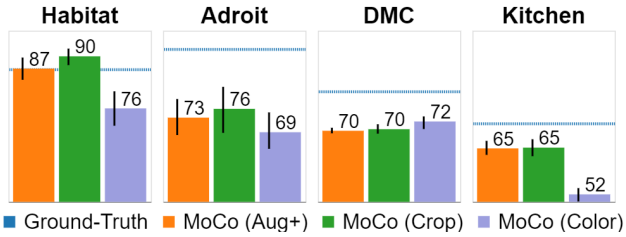

Figure 8: **Invariances comparison in MoCo.** Color augmentation performs worse in all environments except for DMC, while crop augmentation performs the best on average. This suggests that color invariance, commonly used in semantic recognition, is not always suited for control.

resentations. While in-domain pre-training helps compared to training from scratch, it is surprisingly not much better than pre-training on ImageNet or Places. For Habitat, pre-training on Habitat leads to similar performance as pre-training on ImageNet and Places. However, in the case of MuJoCo, PVRs trained on the MuJoCo expert trajectories are not competitive with representations trained on ImageNet or Places. As mentioned earlier, training on larger and more diverse datasets *may* potentially bridge the gap, but is not a pragmatic solution, since we ultimately desire data efficiency in the deployment environment.

This suggests that the key to representations that work on diverse control domains does not lie only in the training dataset. Our next hypothesis is that it perhaps lies in the invariances captured by the model.

4.3. Recognition vs. Control: Two Tales of Invariances

Most off-the-shelf vision models have been designed for semantic recognition. Next, we investigate if representations for control tasks should have different characteristics than representations for semantic recognition. Intuitively, this does seem obvious. For example, semantic recognition requires invariances to poses/viewpoints, but poses provide critical information to action policies. To investigate this aspect, we conduct the following experiment on MoCo. By default, MoCo learns invariances through various data augmentation schemes: crop augmentation provides translation and occlusion invariance, while color jitter augmentation provides illumination and color invariance. In this experiment, we isolate such effects by training MoCo with only one augmentation at a time. In semantic recognition, both color and crop augmentations appear to be critical (Chen et al., 2020). Does this hold true in control as well?

Results in Figure 8 indicate that different augmentations have dramatically different effects in control. In particular, in all domains other than DMC, color-only augmentations significantly under-perform. Furthermore, crop-only augmentations lead to representations that are as good or even better than all other representations. The importance of crop-only augmentations is consistent with prior works as well (Srinivas et al., 2020; Yarats et al., 2021b). We hypothesize that crop augmentations highlight relative displacement between the agent and different objects, as opposed to their absolute spatial locations in the image observation, thus providing a useful inductive bias. Overall, our experiment suggests that control may require a different set of invariances compared to semantic understanding.

4.4. Feature Hierarchies for Control

The previous experiment indicates that invariances for semantic recognition may not be ideal for control. So far, we have leveraged the features obtained at the last layer (after final spatial average pooling) of pre-trained models. This layer is known to encode high-level semantics (Selvaraju et al., 2017; Zeyu et al., 2019). However, control tasks could benefit from access to a low-level representation that encodes spatial information. Furthermore, studies in vision have shown that last layer features are the most invariant and early layer features are less invariant to low-level perturbations (Zeiler & Fergus, 2014), which have resulted in the use of feature pyramids and hierarchies in several vision tasks (Lin et al., 2017). Inspired by these observations, we next investigate the use of early layer features for control. We note that intermediate layers (third, fourth) have more activations than the last layer (fifth). To ease computations and perform fair comparisons, we compress these representations to the size of the representation at the last layer (more details in Appendix A.4). To the best of our knowledge, the use of early layer features is still unexplored in policy learning for control.

Figure 9 shows that early convolution layer features are more effective for fine-grained control tasks (MuJoCo). In fact, they are so effective that they even match or outperform ground-truth features. While the ground-truth state features we use contain complete information –i.e., can function as Markov states– they may not be the ideal representation from a learning viewpoint[2]. Indeed, not only are state features known to impact policy learning performance (Brockman et al., 2016; Ahn et al., 2019), but different representations of the same information –e.g., Euler angles and quaternions– may perform differently (Gaudet & Maida, 2018). At the same time, visual representations may capture higher-level information that makes it easier for the agent to behave optimally.

Furthermore, earlier layer features work better for MuJoCo but not for Habitat. This is perhaps not surprising since navigation in Habitat requires semantic understanding of

[2]We emphasize that the ground-truth features used in our experiments are the default choices provided by the environments and have been used in many prior works.

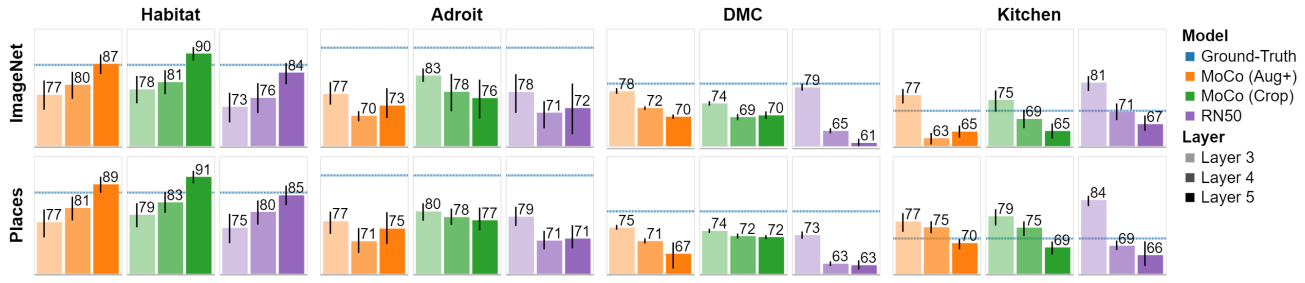

Figure 9: **Success rate when using representations from different layers**. There is a clear trend in Habitat showing that PVRs from later layers (opaque colors) perform better. In contrast, early layer features (transparent colors) perform better in the MuJoCo tasks. The same trends hold across both ImageNet and Places.

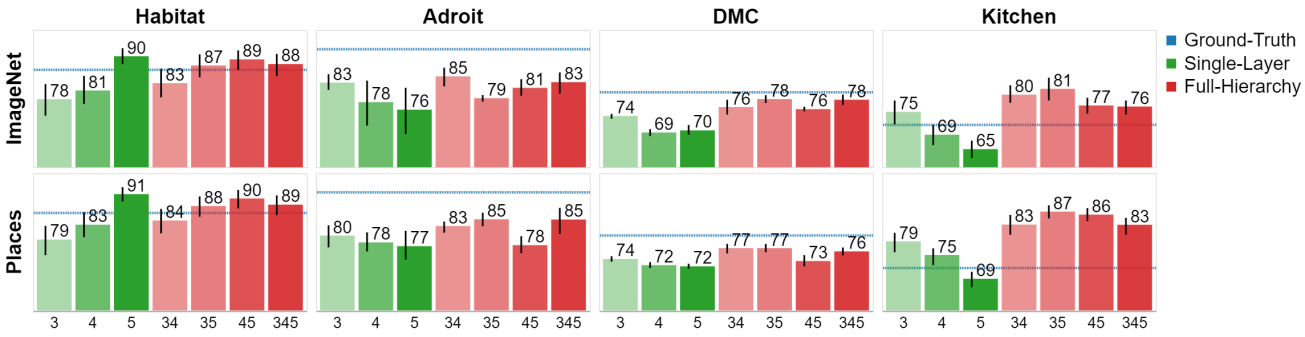

Figure 10: **Single-layer vs. full-hierarchy features of MoCo with crop augmentation.** The latter are competitive in **all** the domains, and in the case of Kitchen even outperform ground-truth features.

the environment. For instance, the agent needs to detect if there is a wall or an obstacle in front of itself to avoid it. This kind of information may be present in the last layer of vision model trained for semantic recognition.

4.5. Full-Hierarchy Models

The experiment in Section 4.4 motivates two new questions. First, can we design PVRs combining features from multiple layers of vision models? Ideally, the policy should learn to use the best features required to solve the task. Second, since PVRs work even when pre-trained on out-of-domain data, could such new full-hierarchy features be "near-universal", i.e., work for any control task –at least those studied here?

Figure 10 shows the performance of PVRs using all combinations of the last three layers of MoCo with crop augmentation, the best model so far. In MuJoCo, any PVR using the third layer features –the best single-layer features– performs competitively with ground-truth features. Similarly, in Habitat any PVR using the fifth layer performs extremely well. This suggests that the policy can indeed exploit the best features from the full-hierarchy to solve the task.

Overall, the PVR using all the three layers (3, 4, 5) performs best on average, and the same PVR is able to solve all the four domains, sometimes even better than ground-truth features. This is an important result, considering that our four control domains are very diverse and span low-level locomotion, dexterous manipulation, and indoor navigation in very diverse environments. Furthermore, this PVR is trained entirely using out-of-domain data and has never seen a single frame from any of these environments. This presents a very promising case for using PVRs for control.

5. Discussion and Conclusion

Freezing vs Fine-Tuning PVR. Our primary motivation in this work was to study the use of representations from pre-trained vision models for control tasks. Consistent with this, our experiments freeze the vision models to directly test the quality of pre-trained representations, and to prevent any "on-the-fly" representation learning. This is similar in spirit to the linear classification (probe) protocol used to evaluate representations in computer vision. Fine-tuning of pre-trained models have been found to be challenging in both computer vision and NLP, especially in the sparse data regime, but may result in marginally improved performance (Hénaff et al., 2020; Peters et al., 2019). Further investigations on fine-tuning, and development of targeted fine-tuning approaches for control, could make for interesting future work.

Imitation Learning vs Reinforcement Learning. In this work, we focused on learning policies using imitation learning (specifically behavior cloning) as opposed to RL. Despite significant advances in learning visuo-motor policies with RL (Yarats et al., 2021a; Wijmans et al., 2020; Hafner et al., 2020), the best algorithms are still data-intensive requiring millions of samples. The use of pre-trained representations are particularly important in the sparse data regime, and thus we choose to train policies with imitation learning. Furthermore, our work required the evaluation of a large collection of pre-trained models across a diverse suite of environments, which was prohibitively expensive with current RL algorithms. We hope that our insights on important considerations for PVRs in the context of control can be used for RL in future work.

Summary of Our Contibutions. The use of off-the-shelf vision models as perception modules for control policies is a relatively new area of research, trying to bridge the gap between advances in computer vision and control. This is a departure from the current dominant paradigm in control, where visual encoders are initialized randomly and trained from scratch using environment interactions.

In this paper, we took a step back and asked fundamental questions about representations and control, in the hope of making a single off-the-shelf vision model –trained on out-of-domain datasets– work for different control tasks. Through extensive experiments, we find that off-the-shelf PVRs trained on completely out-of-domain data can be competitive with ground-truth features for training policies. Overall, we identified three major components that are crucial for successful PVRs. First, SSL models provide better features for control than supervised models. Second, translation and occlusion invariance, provided by crop augmentation, is more relevant for control than other invariances like illumination and color. Third, early convolution layer features are better for fine-grained control tasks (MuJoCo) while later convolution layer features are better for semantic tasks (Habitat).

Towards Universal Representations for Control. Based on these findings, we proposed a novel PVR combining features from multiple layers of a crop-augmented MoCo model trained on out-of-domain data. Our PVR was competitive with or outperformed ground-truth features on all four evaluation domains.

Motivated by these results, we believe that research should focus more on learning control policies directly from visual input using pre-trained perception modules, rather than using hand-designed ground-truth features. While such features may be available in simulation or specialized motion capture systems, they are hard to estimate in unstructured real-world environments. Yet, training an end-to-end visuo-motor policy has difficulties as well. The visual encoders increase the complexity of the policies, and might require a significantly larger amount of training data. In this context, the use of pre-trained vision modules can offer substantial benefits by dramatically reducing the data requirement and improving the policy performance. Furthermore, using a frozen PVR simplifies the control policy architecture and training pipeline.

We hope that the promising results presented in this paper will inspire our research community to focus more on developing a universal representation for control –one single PVR pre-trained on out-of-domain data that can be used as perception module for any control task.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine learning (ICML)*, 2004.

Ahn, M., Zhu, H., Hartikainen, K., Ponte, H., Gupta, A., Levine, S., and Kumar, V. ROBEL: Robotics Benchmarks for Learning with Low-Cost Robots. In *Conference on Robot Learning (CoRL)*, 2019.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym. arXiv:1606.01540, 2016.

Brown, T. B. et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Cao, Z., Simon, T., Wei, S., and Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709, 2020.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor. In *International Conference on Machine Learning (ICML)*. PMLR, 2018.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219, 2020.

Gaudet, C. J. and Maida, A. Deep quaternion networks. *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018.

Girshick, R. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. Scaling and benchmarking self-supervised visual representation learning. In *International Conference on Computer Vision (ICCV)*, 2019.

Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al. Self-supervised pretraining of visual features in the wild. arXiv:2103.01988, 2021.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, 2019.

Hafner, D., Lillicrap, T. P., Ba, J., and Norouzi, M. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Hara, K., Kataoka, H., and Satoh, Y. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum Contrast for Unsupervised Visual Representation Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Heess, N. M. O., Dhruva, T., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S. M. A., Riedmiller, M. A., and Silver, D. Emergence of locomotion behaviours in rich environments. arXiv:1707.02286, 2017.

Hénaff, O. J., Srinivas, A., Fauw, J. D., Razavi, A., Doersch, C., Eslami, S. M. A., and van den Oord, A. Data-Efficient Image Recognition with Contrastive Predictive Coding. arXiv:1905.09272, 2020.

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

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but Effective: CLIP Embeddings for Embodied AI. arXiv:2111.09888, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. 25:1097–1105, 2012.

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

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., and Zaremba, W. Learning Dexterous In-Hand Manipulation. *The International Journal of Robotics Research (IJRR)*, 39(1):3–20, 2020.

Parisi, S., Dean, V., Pathak, D., and Gupta, A. Interesting Object, Curious Agent: Learning Task-Agnostic Exploration. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Transactions on Graphics*, 37:143:1–143:14, 2018.

Peters, M. E., Ruder, S., and Smith, N. A. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. arXiv:1903.05987, 2019.

Purushwalkam, S. and Gupta, A. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

Rafailov, R., Yu, T., Rajeswaran, A., and Finn, C. Visual adversarial imitation learning using variational models. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

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

Rajeswaran, A., Lowrey, K., Todorov, E. V., and Kakade, S. M. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (R:SS)*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

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

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, 2017.

Shah, R. and Kumar, V. RRL: ResNet as representation for Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Shang, W., Wang, X., Srinivas, A., Rajeswaran, A., Gao, Y., Abbeel, P., and Laskin, M. Reinforcement Learning with Latent Flow. In *Advances in Neural Information Processing Systems (NIPS)*, 2021.

Srinivas, A., Laskin, M., and Abbeel, P. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2020.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021.

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

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision (ICCV)*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, March 1998.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. DeepMind Control Suite. arXiv:1801.00690, 2018.

Tieleman, T. and Hinton, G. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical Report*, 2017.

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

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, 2016.

Weihs, L., Deitke, M., Kembhavi, A., and Mottaghi, R. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Wijmans, E., Kadian, A., Morcos, A. S., Lee, S., Essa, I., Parikh, D., Savva, M., and Batra, D. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Yarats, D., Kostrikov, I., and Fergus, R. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In *International Conference on Learning Representations (ICLR)*, 2021b.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

Zeyu, F., Xu, C., and Tao, D. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

A. Training Details

A.1. Habitat Details

Visual Input. PVR models are fed with two 64×64 RGB images, one for the view of the scene from the agent's perspective, and one for the target location. Each image is encoded independently by the model, and the two encodings are concatenated before being passed to the policy.

Policy Architecture. The PVR passes through a batch normalization layer and then through a 2-layer MLP (ReLU activation), followed by a 2-layer LSTM and then a 1-layer MLP (softmax activation). All hidden layers have 1,024 units. Ground-truth features do not use batch-normalization, as it significantly harmed the performance.

Policy Optimization. Following Parisi et al. (2021), we update the policy with 16 mini-batches of 100 consecutive steps with the RMSProp optimizer (Tieleman & Hinton, 2017) (learning rate 0.0001). Gradients are clipped to have max norm 40. Learning lasts for 125,000 policy updates.

Imitation Learning Data. We collect 50,000 optimal trajectories (10,000 per scene) using Habitat's native solver, for a total of ∼2,100,000 samples.

Success Rate. The policy success rate is estimated over 50 trajectories, and further averaged over the last six policy updates, for a total of 300 trajectories per seed.

A.2. MuJoCo Details

Visual Input. Consistent with prior works, the visual input takes the last three 256×256 RGB image observations of the environment. Each image is encoded independently by the PVR model. These three PVRs are fused together by using latent differences following the work of Shang et al. (2021). We *do not* use any other proprioceptive observations like joint encoders for hands, and our policies are based solely on embeddings of the visual inputs.

Policy Architecture. The fused PVR passes through a batch normalization layer and then through a 3-layer MLP with 256 hidden units each and ReLU activation.

Policy Optimization. We update the policy with mini-batches of 256 samples for 100 epochs with the Adam optimizer (Kingma & Ba, 2014) (learning rate 0.001).

Imitation Learning Data. We collect trajectories using an optimal policy trained with RL (Rajeswaran et al., 2017; 2018). The amount of data depends on the task difficulty.

- Adroit: 100 trajectories per task with 100- and 200-step horizon for `Reorient Pen` and `Relocate`, respectively. The total number of samples is 30,000.
- DeepMind Control: 100 trajectories per task. We use an action repeat of 2, resulting in a 500-step horizon per trajectory. The total number of samples is 250,000.
- Franka Kitchen: 25 trajectories per task with 50-step horizon for all tasks. The total number of samples is 25,000.

Success Rate. We evaluate the policy every two epochs over 100 trajectories, and report the average performance over the three best epochs over the course of learning. This way we ensure that each representation is given sufficient time to learn, and that the best performance is reported.

A.3. PVRs Details

Datasets

- ImageNet: 1.2 million images.
- Places: 1.8 million images.
- Habitat: ∼2.4 million images. We collect 20,000 optimal trajectories from all the 18 Replica scenes, keeping only one frame every three for the sake of diversity.
- MuJoCo: we use the same aforementioned trajectories, for a total of 30,000 (Adroit), 250,000 (DeepMind Control) and 25,000 (Kitchen) images.

Vision Models

- ResNet: `github.com/pytorch/vision`.
- MoCo: `github.com/facebookresearch/moco` (v2 version).
- CLIP: `github.com/openai/CLIP` (ViT-B/32 and RN50 versions).

A.4. Intermediate Layers Compression

In Section 4.4 we discussed the use of features from intermediate layers of vision models. However, the number of activations in these layers (third, fourth) is significantly higher compared to the representation at the last layer (fifth). To avoid prohibitively expensive compute requirements and perform fair comparisons across layers, we compress these representations to a common size, i.e., the size of the representation at the fifth layer. This is accomplished by adding two residual blocks to the model at the chosen intermediate layer. Similar to an autoencoder model, the first residual block compresses the number of channels, while the second residual block expands the number of channels back to the original. With these additional layers randomly initialized, the model is fine-tuned on the original pre-training task. The output of the first residual block provides the compressed features which are then used in our experiments.

A.5. Compute Details

Vision models pre-training and layer compression was distributed over two nodes of a SLURM-based cluster. Each node used four NVIDIA GeForce GTX 1080 Ti GPUs. Pre-training one PVR model took between 1-3 days depending on the training method, size of the model, and dataset used. Policy imitation learning was performed on a SLURM-based cluster, using a NVIDIA Quadro GP100 GPU. Training one policy took between 8-24 hours (including policy evaluation) depending on the PVR and the environment.