2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SpawnNet: Learning Generalizable Visuomotor Skills from Pre-trained Network

Xingyu Lin* [1] John So* Sashwat Mahalingam Fangchen Liu Pieter Abbeel
UC Berkeley

***Abstract*— The existing internet-scale image and video datasets cover a wide range of everyday objects and tasks, bringing the potential of learning policies that generalize in diverse scenarios. Prior works have explored visual pre-training with different self-supervised objectives. Still, the generalization capabilities of the learned policies and the advantages over well-tuned baselines remain unclear from prior studies. In this work, we present a focused study of the generalization capabilities of the pre-trained visual representations at the categorical level. We identify the key bottleneck in using a frozen pre-trained visual backbone for policy learning and then propose SpawnNet, a novel two-stream architecture that learns to fuse pre-trained multi-layer representations into a separate network to learn a robust policy. Through extensive simulated and real experiments, we show significantly better categorical generalization compared to prior approaches in imitation learning settings. Open-sourced code and videos can be found on our website: `https://xingyu-lin.github.io/spawnnet/`.**

I. Introduction

To take steps towards deploying robots in our daily lives, learning skills that can handle diverse situations is crucial for robots to subsist with us in the real world. For example, consider a scenario where we want our robot to help us wear hats. We can teach this skill by demonstrating the robot with the hats we currently have. However, we don't want to recollect demonstrations once we buy a new hat or change the position of our coat rack. Thus, we hope to enable robots with a smart hat-wearing policy that can handle object variations in color, pose, and shape.

To this end, we need to equip manipulation skills with semantic knowledge of the world, which already lies in current internet-scale video and image datasets [8], [11], [30]. Given the success of visual representation learning [16], [12], [15], [4], [7], [24] on a wide range of computer vision tasks, a natural solution to our goal is to take the pre-trained representations out of the box and train robust visuomotor skills on top of it. Prior works have explored this direction [27], [23], [21], [19], with promising results on improved sample efficiency and asymptotic performance. However, most evaluations in prior works are done on simple tasks with relatively small variations and thus are not sufficiently difficult. Indeed, recent work [14] has shown that a carefully done baseline that learns from scratch can be very competitive with these prior works that use pre-trained networks.

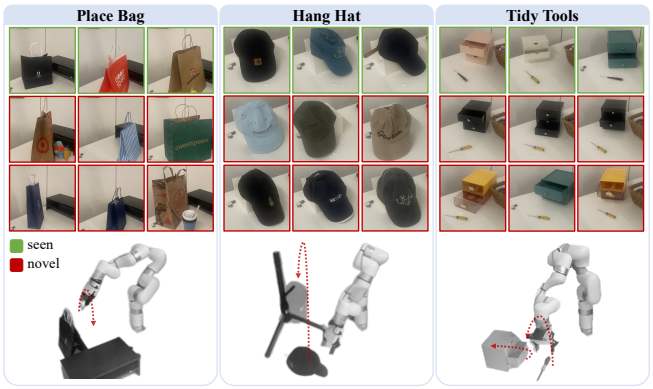


Fig. 1: We consider three challenging categorical manipulation tasks in the real world. For each task, we train on three instances (green boxes) and test on held-out instances (red boxes), with additional variations in poses, articulation, visual distraction, and deformation.

We hypothesize that the benefits of pre-training will be more pronounced in more complex generalization tests, where a trained policy is evaluated for the ability to adapt its behavior to unseen objects. Hence, we create three challenging categorical manipulation tasks (depicted in Figure 1). Our tasks contain a set of diverse objects with no overlapping instances between training and evaluation. Through these challenging tasks, we identify the bottleneck in the existing methods: frozen pre-trained networks give fixed representations to the policy and as such, may hinder policy learning without adapting the visual backbone. The need for adaptation may come from the differences between the pre-training and policy learning objectives, or from the distribution shift from the pre-training dataset to robotic demonstrations, including domains, tasks, or camera viewpoints [35].

Inspired by the ProgressiveNet [29] in the continual learning literature, we propose a simple two-stream architecture, SpawnNet, to learn a generalizable visuomotor policy from a pre-trained neural network. Instead of directly using frozen representations, we adopt a separate network that learns to fuse multi-layer pre-trained representations (Figure 2) and generate actions. The learnable stream incorporates domain-specific features from the raw observations to help handle distribution shifts. Meanwhile, it can take advantage of the pre-trained features at different layers for faster learning and better generalization. While other transfer learning methods

[1] * Indicates equal contribution. All authors are from UC Berkeley. Correspondence to Xingyu Lin: `xingyu@berkeley.edu`

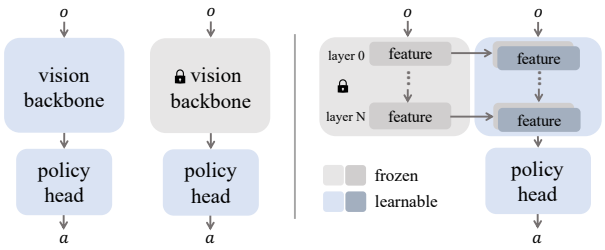
have shown successes in CV and NLP [17], [18], [5], including simple fine-tuning, they can underperform in our case due to the large discrepancy between visual pre-training and downstream control tasks. Through extensive experiments, we demonstrate that SpawnNet significantly outperforms prior approaches and a learning-from-scratch method when tested on held-out novel objects. Below we highlight the contribution of this paper:

1) We propose SpawnNet, a novel and flexible framework that can adapt any pre-trained model to a generalizable visuomotor policy on downstream manipulation tasks.
2) We perform systematic evaluations of different methods utilizing pre-trained representation both in simulation and in the real world, showing significant improvement of our method in cross-instance generalization.

II. Related Works

Visual Pre-training for End-to-end Policy Learning. Recently there has been a surge of works that aim to pre-train a visual representation on image or video datasets, using ResNet [32], [25], [23] or Vision Transformer (ViT) [27], [20] as the backbone. After training, these works freeze the visual representation for downstream policy learning. It has been shown that pre-trained representation can achieve better sample efficiency or asymptotic performance. However, evaluations in prior works are done in environments with limited visual variations, such as changes in object poses. Some works study changes in backgrounds [6], [34], which do not require the policy to adapt its action sequences. As such, a recent study shows that strong learning-from-scratch baselines with data augmentations can achieve comparable performance to the best pre-trained representations [14]. In contrast, our work sheds light on the generalization capabilities of pre-trained representations on a rich set of assets and tasks in both simulation and real-world experiments through a novel neural architecture that better utilizes the pre-trained representation. We note concurrent work that uses pre-trained networks in large-scale manipulation tasks [2], while our study is more focused and studies the challenging categorical manipulation tasks.

Transfer Learning. Transferring knowledge from a pre-trained network has led to success in many downstream tasks in CV and NLP, using a frozen representation or fine-tuning as the de facto approach during adaptation. As the pre-trained networks become larger, parameter-efficient fine-tuning approaches (PEFT) have also been proposed by inserting adaptation modules in the pre-trained network [17], [26], [18], [5]. However, few works have explored adaptation architectures for vision-based control in robotics. Sharma et al. [33] apply the PEFT idea for adapting a pre-trained representation. However, PEFT methods may not have enough capacity to overcome the gap between the visual pre-training and downstream control tasks, as we will show in experiments. Our architecture is similar to Progressive Neural Network (ProgNet) [29] in continual learning, but extends over ProgNet by having asymmetric network architectures for the pre-trained stream and control stream.



(a) From scratch (b) From pre-trained (c) SpawnNet

Fig. 2: Prior approaches for learning policies (a) from scratch, (b) from a pre-trained visual representation with a frozen backbone, and (c) the proposed two-stream architecture. The right figure (d) shows their performances on a real-world imitation learning task, evaluated on both seen and unseen instances in a category.

III. Method

Pre-trained visual representations are trained on large image and video datasets and have the potential to help learn generalizable visuomotor skills. We will first give a background on the vision transformer, the pre-trained vision backbone architecture we use in Section III-A. Then in Section III-B, we explain the issue with the current way of using the pre-trained features and present our architecture. Finally, we explain our policy learning frameworks for evaluation in Section III-C.

A. Vision Transformer Preliminary

While our method can be combined with any pre-trained visual architecture, we will mainly discuss its instantiation with pre-trained vision transformers [9] as they are more scalable and commonly used for visual pre-training. The input RGB images are first split into patches. Each patch is encoded into a latent vector representation named a token. Additionally, a learned CLS token is concatenated to the other image tokens, and the sequence is passed through multiple transformer layers. Each transformer layer consists of alternating layers of multi-headed self-attention (MHSA) and MLPs, with residual layers between them. Within the MHSA block, an input X is first split, then projected onto learned key, query, and value bases before the attention mechanism: $MHSA(X) = softmax(QK^T/\sqrt{d})V$, where Q, K, V are linear projections of X.

Prior works in using the pre-trained representation for policy learning usually take the CLS token at the last layer as the image representation. On the other hand, people have found the attention features in MHSA in intermediate layers of the vision transformer to encode semantic information at the object-level and part-level [1], [13]. As such, we follow [1] to extract the key K in MHSA from a pre-trained vision transformer DINO [3]. Since vision transformers usually use a large model and require large computation to fine-tune, throughout this paper we freeze the weights of the pre-trained network.

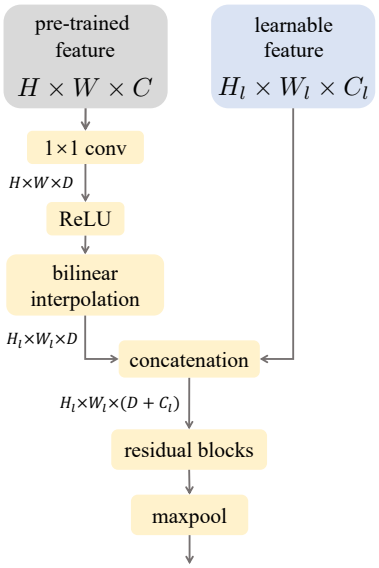


Fig. 3: Adapter layers to fuse the pre-trained features with the learnable features.

B. SpawnNet Architecture

Given these powerful pre-trained vision transformers, how do we utilize them for policy learning? Prior approaches extract the CLS token at the last layer as the image representation, as illustrated in Figure 2. The representation is typically frozen as fine-tuning vision transformers requires large computation. However, this architecture lacks flexibility in adapting the visual features. This potentially hinders downstream policy learning for two reasons. First, the self-supervised objectives in pre-training may not align with the policy learning objective. Second, datasets used for pre-training can have a different distribution from downstream tasks in domains, tasks, or camera viewpoints [35].

To learn more flexible task-specific representations while taking advantage of the pre-trained features, SpawnNet trains a separate stream of convolutional neural networks (CNN) from scratch, taking the raw observation as input (Figure 2). At the same time, we design adapter modules to fuse the new stream with the pre-trained features at different layers, taking advantage of the robust features as needed. We find that a shallow CNN for the new stream works well under the low-data regime during policy learning. Additionally, this architecture allows us to incorporate different modalities like depth in the separate stream, which is not in the pre-trained vision networks.

We first extract spatially dense descriptors from different layers of the pre-trained networks. At the l^{th} layer of the vision transformer, we extract the key feature in MHSA from each token, forming a feature grid of $\phi^l(I) \in \mathbb{R}^{H \times W \times C}$, where $H \times W$ is the number of image patches and C is the feature dimension. Extracting features from different layers allows us to extract both low-level and high-level visual features from the pre-trained network [1]. Note that vision transformers keep the same number of image patches at each layer; extracting features from each image patch allows us to maximally preserve spatial information.

Adapter Layers. We fuse the features from the two streams of networks at different layers as shown in Figure 3 using *adapter layers*. The adapter first maps the pre-trained feature into a latent embedding of size D through a convolutional kernel of size 1 and stride 1. It then resizes the feature from $H \times W$ to $H_l \times W_l$ with bi-linear interpolation and finally concatenates them in the feature dimension. It is then passed through two residual blocks to be processed in the next layers. We note that the design of the adapter layers allows us to utilize flexible architectures for downstream control tasks. This is a key difference to ProgressiveNet [29] and enables us to effectively learn policies from limited demonstration data. We will demonstrate this in the experiments.

C. Visuomotor Policy Learning

We study the effect of visual representation in learning challenging visuomotor control tasks. For all our tasks, we consider large intra-category variations and hold out a portion of instances during training to study generalization.

Learning with Expert Guidance in Simulation. Prior works study the visual representations under the Reinforcement Learning (RL) or Behaviour Cloning (BC) framework. This couples the visual representation with challenges in exploration or covariate shift. For our simulation experiments, we first train RL policies on all training instances using PPO [31]. We treat the RL policies as experts and learn image-based policies using DAgger [28]: Within each iteration, we first roll out the current policy in the environments and then query the experts on agent's visited states to get the corresponding action labels for training. We then train the current policy with gradient descent updates to minimize the MSE loss between the agent's actions and the expert's actions.

Behavior Cloning with Teleoperation Demonstration in the Real World. While the simulated tasks allow us to iterate quickly, the rendered images are not photo-realistic, which creates an artificial gap to the real data used for pre-training. As such, we further train and evaluate visuomotor policies in the real world. For each task, we collect demonstrations from humans using a teleoperation setup. Since it is difficult to query humans in robot-visited states in the real world, we simply train different policies using behavior cloning.

IV. Experiments

Below, we provide experiments for (1) comparisons with prior works using pre-trained representations in simulation (Sec. IV-A and the real world (Sec. IV-B (2) visualization of the emergent attention on parts (Sec. III-C) (3) ablation study (Sec. IV-D) (4) SpawnNet with different pre-trained networks (Sec. IV-E and (5) comparisons with other transfer learning methods (Sec. IV-F). We note that experiments in prior works ([27], [23], [14]) have only shown limited variations in poses or instances within a category. We aim to stress-test the generalization capabilities of the policies

through significant in-category variations where a policy would break if not well adapted.

Baselines. We compare with the following baselines:

- **LfS+aug (Learning from Scratch) [14]** is a shallow ConvNet encoder followed by an MLP policy with data augmentation, which has been shown to be a strong baseline. Following [14], we use random shift as our data augmentation.
- **MVP [27]** trains a masked auto-encoder from egocentric video data and takes the frozen CLS token from the vision transformer as the representation.
- **R3M [23]** trains a language-aligned visual representation from videos using time contrastive learning and language-video alignment.

All methods with only a frozen pre-trained backbone (i.e. **MVP**, **R3M**) take RGB as input. Meanwhile, incorporating depth is straightforward for **SpawnNet** and **LfS** by adding a depth channel to the trainable encoder. We denote these depth-conditioned variants as **SpawnNet+d** and **LfS+aug+d**.

A. Simulation Experiments

Tasks. We conduct experiments on two tasks in simulation, opening different cabinet doors and drawers with a Franka arm, as shown in Figure 4. For each task, we hold out a subset of the objects during training and use them for evaluating the generalization of the learned policies on novel objects. Our tasks are taken from RLAfford [10], built on top of IsaacGym [22]. The agents receive image observations from three camera views (left, middle, right) and the proprioception as input, and output joint positions for the arm.

Training and Evaluation. We first take the pre-trained RL policies from [10] using PPO [31], which takes point clouds as input. We then train different policies with DAgger to imitate the RL experts. This helps us decouple the visual representation from other difficulties such as the covariant shift in behavior cloning. We alternate model training and agent rollouts and stop after a fixed number of trajectories. For all methods, we select 21 instances from the training set where the experts perform well and use them for training DAgger policies. We evaluate policies on 8 held-out instances for Open Door and 12 held-out instances for Open Drawer. We roll out the agent for 5 trajectories per asset to evaluate each model. For each method, we train three models with different seeds. We report the average performance and the standard error across training and novel instances.

The results are shown in Figure 4. We find that different approaches perform similarly in training instances, while SpawnNet generalizes better than other pre-trained representations. LfS+aug proves to be a strong baseline in simulation. We next evaluate in a real-world setting, where visual pre-training should comparatively benefit from realistic image observations and limited training instances.

B. Real World Experiments

Experimental Setup. We evaluate methods using a real-world robotics setup with a UFACTORY xArm 7. Humans provide demonstrations using a 3Dconnexion Spacemouse; actions are parameterized and collected as 6-DOF, delta end-effector control $(\Delta x, \Delta y, \Delta z, \Delta\alpha, \Delta\beta, \Delta\gamma)$ at 5Hz, plus gripper open/close. Both translation and rotation in the action space are defined in the wrist camera view to enable better generalization. Each demonstration collects observations as RGB and depth images from two RealSense cameras: one from third-person view, and one attached to the wrist of the robot arm. We stack the most recent four frames as the agents' observations. We do not give the agents access to proprioception information as we find that the agent tends to overfit to the proprioception and ignore the visual input, which hinders generalization.

For real-world comparison, we evaluate against **R3M** and **LfS+aug+d** as they perform well in simulation tasks.

Tasks. We evaluate methods with behavior cloning for three manipulation tasks. Each real-world task consists of around 90 demonstrations across 3 training instances with pose variations and a set of held-out objects. The tasks and variations are illustrated in Figure 1 and summarized below:

1) **Place Bag**: Lift up a bag by the strap, and place it on a table. The bag's pose is varied.
2) **Hang Hat**: Pick up a cap, and hang it on a rack. Caps' poses slightly vary across runs.
3) **Tidy Tools**: Pick up a handled tool, place it in an open drawer, and close the drawer. We provide different tools, different initial tool poses, different drawers, and drawers to close.

The combination of geometrically and visually diverse instance variation makes both learning manipulation skills and generalizing to new instances challenging.

Results. The results on training and held-out instances are shown in Figure 5. Numeric values, training hyperparameters, and more details can be found in the Appendix[1]. We observe that SpawnNet performs much better than baselines in both training and unseen instances; in unseen instances, the overall gap over the baselines becomes even larger than in simulation. We conjecture that this is because real-world images follow a closer distribution to the pre-training dataset compared to simulation. While LfS and R3M perform well on training occasionally, they are much worse at generalizing. For tasks with larger instance variations on training, such as Tidy Tools, both LfS and R3M fail to achieve 50% success rates even on training while SpawnNet achieves greater than 80% success. We hypothesize that this is because SpawnNet can better capture semantic variance across demonstrations on different instances through dense features. Additionally, by adding depth, SpawnNet+d improves by as much as 15% over SpawnNet on novel instances. This shows the flexibility of SpawnNet in incorporating different modalities.

C. What does SpawnNet learn from the pre-trained network?

We provide more insights into how SpawnNet's use of pre-trained features can help policy learning; to do so, we

[1]Appendix link: https://xingyu-lin.github.io/spawnnet/files/appendix.pdf

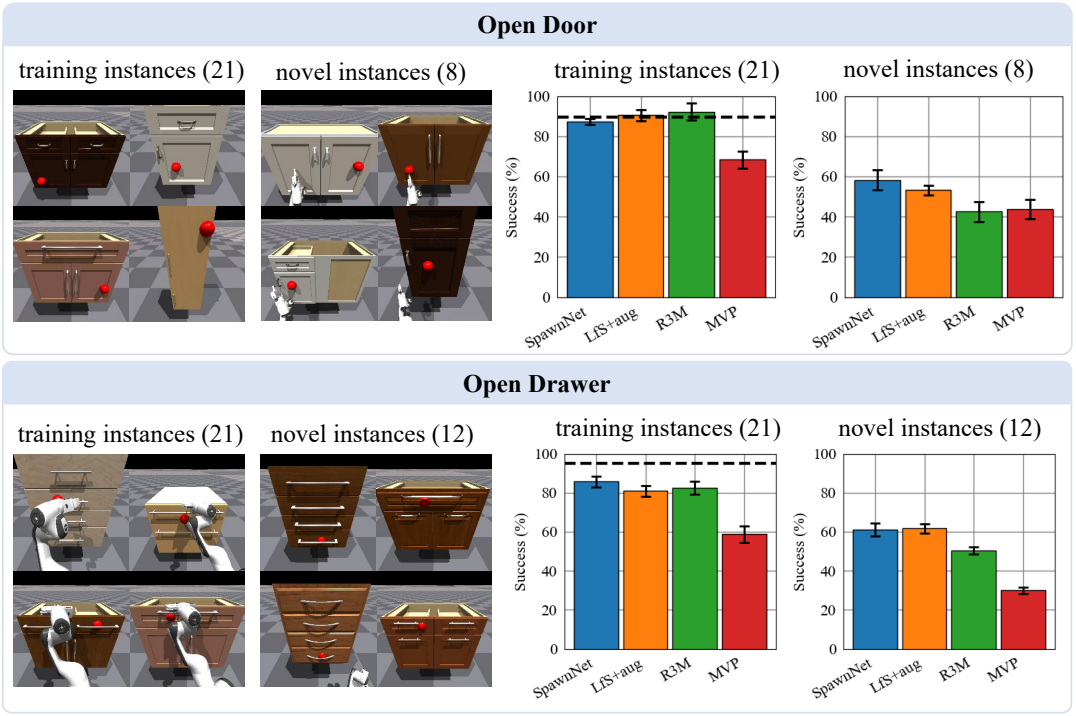


Fig. 4: Simulation results on Open Door and Open Drawers. The left figures show the training and novel instances we use. The observations are rendered from the agent's middle camera. We add red spheres in the scene to specify the task of which door/drawer to open. The right shows success rates of different methods in both seen and unseen instances after a fixed number of agent rollouts. The dashed black line shows the RL expert's performance. The error bars show the standard error computed from three random seeds. Numbers in the brackets denote the number of instances.

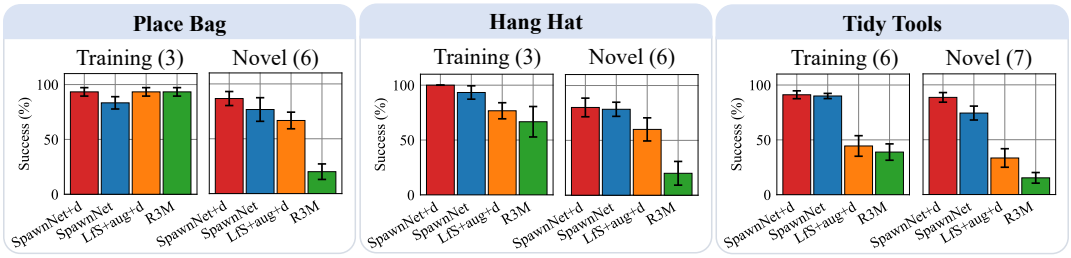


Fig. 5: Real-world manipulation results on three tasks. All methods here use the same data augmentation. We evaluate each method on each instance over 5 trials (more than 30 trials on novel instances). We report the mean and standard error.

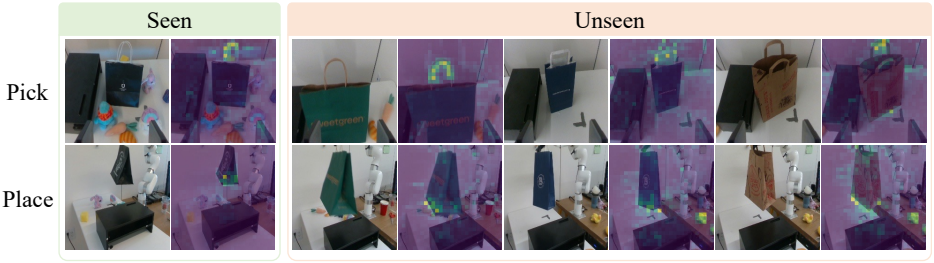


Fig. 6: Visualization of the attention on the pre-trained features by the last adapter during the *Place Bag* task. The attention focuses on important regions that are consistent across even unseen instances, e.g. the bottom of the bags are highlighted when placing.

take a trained policy and visualize the norm of the features after the 1x1 convolutions and the nonlinear layers in the last layer of the adapter, as shown in Figure 6. We see that important parts of the bags are highlighted, such as the handles of the bags when picking and the bottom of the bags when placing. These highlighted regions are consistent across time and across instances. They also generalize to unseen instances. As such, we believe that fusing the pre-trained network's layers enables better learning and generalization. Please refer to the website for more visualizations.

TABLE I: Performance Comparison

Tasks	SpawnNet	Frozen	Full Finetune	Last-layer Finetune	LORA Finetune	ProgressiveNet
Place Bag	**86.7 ± 6.4**	11.7 ± 4.4	8.3 ± 4.4	15.0 ± 5.3	28.3 ± 6.4	51.7 ± 8.3
Tidy Tool	**88.6 ± 4.4**	7.6 ± 2.8	-	26.7 ± 4.5	41.9 ± 7.0	43.8 ± 6.1

D. Ablations

We ablate SpawnNet's architecture using two tasks, Open Door (sim) and Place Bag (real):

- **Remove pre-trained features (-Pre-trained)**: To study how much we gain from the pre-trained representation, we zero-mask all pre-trained features.
- **Last pre-trained layer only (-Multiple)**: To study the importance of features from multiple layers, we only adapt the last pre-trained layer's features.
- **CLS token only (-Dense)**: To study the importance of dense features, we replace pre-trained layers' dense features [HxWxC] with the CLS token [1xC] tiled to the same dimensions.

Method	Open Door (Sim)	Place Bag (Real)
Full Method	59.6	86.7
-Pre-trained	52.9 (-6.7)	-
-Multiple	58.3 (-1.3)	73.3 (-13.3)
-Dense	52.5 (-7.1)	61.7 (-24.9)

TABLE II: Ablations on Open Door and Place Bag; numbers in red indicate performance decrease compared to the full method. Removing dense features has the most impact.

Results are shown in Table II; **Full Method** denotes SpawnNet+d. Notably, both removing dense features and pre-trained features entirely hurt performance, suggesting the importance of dense features for generalization. Adapting only the last pre-trained layer results in a slight decrease in performance, suggesting that spatial information from multiple layers is also helpful.

E. SpawnNet with other pre-trained networks

Method	Open Door	Place Bag
DINO	44.6	11.7
R3M	42.5	20.0
MVP	43.8	-
SpawnDINO	58.3 (+13.7)	86.7 (+75.0)
SpawnR3M	49.2 (+6.7)	81.7 (+61.7)
SpawnMVP	53.3 (+9.5)	-

TABLE III: Performance on the Open Door and Place Bag tasks with different pre-trained networks. The numbers in green show improvement over the pre-trained network. Using SpawnNet improves the performance of all pre-trained representations.

We additionally experiment with initializing SpawnNet from other pre-trained backbones (**SpawnDINO**, **SpawnMVP**, and **SpawnR3M**), and compare them to pre-trained networks without SpawnNet (**DINO**, **MVP**, and **R3M**). These models vary broadly in terms of dataset, training objectives, and architecture. Again, we evaluate using the simulated Open Door and real Place Bag tasks. The results in Table III show SpawnNet is a general architecture that can improve the performance of any pre-trained network.

F. Comparison with Transfer Learning Methods

In this section, we additionally compare with other transfer learning methods. The results are shown in Table I. Here, the experiments are performed over the DINO pre-trained ViT for two real-world tasks. SpawnNet significantly outperforms other transfer learning methods across the board. Fine-tuning methods do not perform as well as SpawnNet, showing the difficulty of fine-tuning transformer models with limited demonstration data. Moreover, SpawnNet outperforms ProgressiveNet. The main difference is that SpawnNet uses a CNN for the control stream to take advantage of the inductive bias given limited demonstration data while ProgressiveNet uses symmetric architectures (i.e. ViT).

V. Limitations

Our experiments mainly study policy generalization under imitation learning settings, which requires the action distributions to be similar in training and evaluation tasks. For this reason, we only evaluate the policy on novel instances, without heavily extrapolating the pose of instances (see Appendix C for details), as an out-of-distribution pose can cause a drastic change of motion. However, this can be addressed in an interactive learning setting. We leave incorporation with reinforcement learning algorithms to future work.

VI. Conclusions

In this paper, we propose a novel and effective architecture to take advantage of pre-trained representations beyond simply freezing the representation. We show that SpawnNet's use of the pre-trained representation not only improves performance on training instances, but more importantly, offers better generalization to novel instances when compared to competitive learning-from-scratch and pre-trained baselines. Through systematic evaluation, we hope our paper can convince more people of the benefits of using pre-trained visual representations and boost further progress.

ACKNOWLEDGMENT

We would like to thank Youngwoon Lee, Xingyang Geng, Boyi Li, Qi Dou, and Jason Anderson for the fruitful discussions. We also thank Olivia Watkins, Yuqing Du and Oleh Rybkin for their feedback on the initial draft of the paper. This work was supported in part by NSF under the AI4OPT AI Center, the BAIR Industrial Consortium, and the InnoHK of the Government of the Hong Kong Special Administrative Region via the Hong Kong Centre for Logistics Robotics.

REFERENCES

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCV Workshop on What is Motion For?*, 2022.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[5] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *International Conference on Learning Representations (ICLR)*, 2023.

[6] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. In *Robotics: Science and Systems (RSS)*, 2023.

[7] Davide Chicco. Siamese neural networks: An overview. *Artificial neural networks*, pages 73–94, 2021.

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

[10] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. *International Conference on Robotics and Automation (ICRA)*, 2023.

[11] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[13] Denis Hadjivelichkov, Sicelukwanda Zwane, Lourdes Agapito, Marc Peter Deisenroth, and Dimitrios Kanoulas. One-shot transfer of affordance regions? affcorrs! In *Conference on Robot Learning (CoRL)*, 2022.

[14] Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *ICML*, 2023.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[17] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, 2019.

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

[19] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[20] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[21] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

[22] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[23] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[25] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning (ICML)*, 2022.

[26] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

[27] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning (CoRL)*, 2022.

[28] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011.

[29] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

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

[32] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021.

[33] Mohit Sharma, Claudio Fantacci, Yuxiang Zhou, Skanda Koppula, Nicolas Heess, Jon Scholz, and Yusuf Aytar. Lossless adaptation of pretrained vision models for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2023.

[34] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[35] Tony Zhao, Siddharth Karamchetim, Thomas Kollar, Chelsea Finn, and Percy Liang. What makes representation learning from videos hard for control? In *RSS 2022 Workshop on Scaling Robot Learning*, 2022.