Playful Interactions for Representation Learning

Sarah Young UC Berkeley
Jyothish Pari NYU
Pieter Abbeel UC Berkeley
Lerrel Pinto NYU

Abstract

One of the key challenges in visual imitation learning is collecting large amounts of expert demonstrations for a given task. While methods for collecting human demonstrations are becoming easier with teleoperation methods and the use of low-cost assistive tools, we often still require 100-1000 demonstrations for every task to learn a visual representation and policy. To address this, we turn to an alternate form of data that does not require task-specific demonstrations – *play*. Playing is a fundamental method children use to learn a set of skills and behaviors and visual representations in early learning. Importantly, play data is diverse, task-agnostic, and relatively cheap to obtain. In this work, we propose to use playful interactions in a self-supervised manner to learn visual representations for downstream tasks. We collect 2 hours of playful data in 19 diverse environments and use self-predictive learning to extract visual representations. Given these representations, we train policies using imitation learning for two downstream tasks: *Pushing* and *Stacking*. Our representations, which are trained from scratch, compare favorably against ImageNet pretrained representations. Finally, we provide an experimental analysis on the effects of different pretraining modes on downstream task learning. Playful interaction data and models are publicly available on our project website [1].

1 Introduction

Imitation learning has proven to be a powerful approach to learn complex robotic skills from visual observations [1, 2, 3, 4]. Recent works have shown how simple approaches like behavior cloning can reliably replicate manipulation behaviors [5, 6, 7]. However, such methods are notoriously data hungry, often requiring 100-1000 demonstrations during training. This paradigm of visual imitation becomes even less practical when we need to learn a multitude of diverse skills for our robots.

But why does visual imitation require such large amounts of data? One hypothesis is that the imitated policy not only needs to learn the desired behavior, but also the appropriate low-dimensional representation for the high-dimensional visual inputs. Hence one path to efficient visual imitation is to reduce the burden of representation learning by using pretrained representation learning models. In the context of robotics, obtaining reliable pretraining is not straightforward. Standard vision datasets [8, 9, 10] contain various object-centric biases while standard robotic datasets [11, 12] are often lab-specific and contain their own robot-specific biases. This brings us to our central question – How can we get data that matches the visual distribution of a given robot?

To answer this, we take inspiration from research in human development and look at an alternate form of data: *play* [13, 14, 15, 16, 17]. From a pure data perspective, playful interactions possess two key qualities. First, it would be cheap to obtain since play is task-agnostic, and it does not need extensive curation or instruction to data collectors. Second, it would be naturally diverse since playful interactions can be easily collected in unstructured environments.

In this work, we present a framework for representation learning that can scalably collect and learn from playful interactions. First, we use reacher-grabber tools [18] built on top of DemoAT [19]

[1] https://sarahisyoung.github.io/play.html

Physical Reasoning and Inductive Biases for the Real World at NeurIPS 2021.

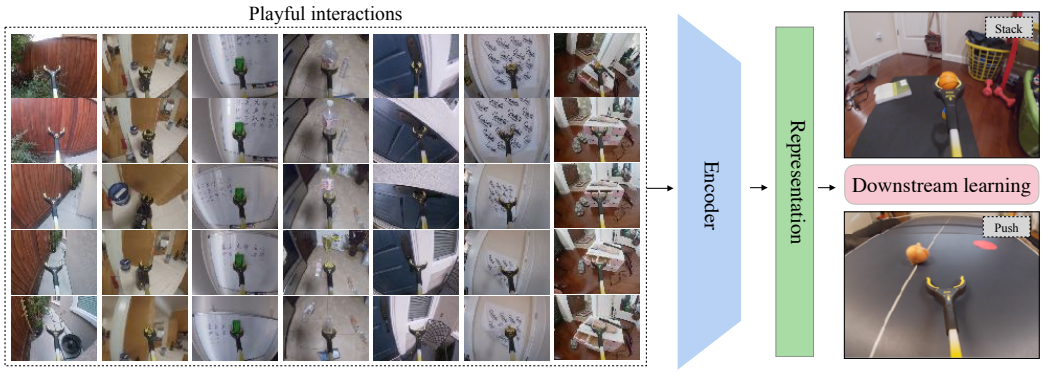

Figure 1: Our method uses two hours of unlabeled, diverse, and unstructured playful interaction data to learn meaningful representations for downstream manipulation tasks such as Pushing and Stacking.

to collect two hours of diverse, self-guided play data in the wild. Equipped with this data, we then use a novel self-supervised learning approach to learn a visual encoder that can extract visual representations. On two downstream tasks, pushing and stacking, we report significant improvements in behavior cloning metrics and outperform popular methods such as imitation from scratch [20], data augmentation based imitation [19], ImageNet based pretraining and multi-task transfer [21, 22]. Against our strongest baseline, ImageNet pretraining, we show that play pretraining achieves up to 27% better MSE performance during test time. When pretrained on top of ImageNet initialization, we achieve up to 38% better performance than training from scratch.

In summary, we present three contributions in this work. First, we propose a framework for collecting playful visual interaction data in the wild. Second, we use self-prediction based representation learning to learn meaningful task-agnostic visual representations. Third, we show that our representations learned on around 2 hours of play can outperform standard imitation-based approaches on two manipulation tasks, pushing and stacking. Although the use of play data has been previously explored in the context of simulated environments [23], to our knowledge this work is the first that studies the use of this play data in real-world environments.

2 Approach

Playful Interactions: We define “playful interactions" as interactions of any kind in a real-world environment using the DemoAT [19] framework. We asked four people to collect data, and these users were untrained and given no information about the downstream tasks. The only guideline we gave data collectors was to “walk around with the reacher-grabber tool and do whatever you want". This includes walking and exploring the space, placing objects, as well as accidental drops. This style of data is very different from our task-specific data, which only consists of expert, goal-oriented trajectories. This kind of unstructured data is useful because it contains exploratory and sub-optimal behaviors that are critical to learning generalizable and robust representations. More importantly, it is much easier to obtain. Data collection can be done by any individual, even young children, and existing data collected for other purposes can also serve as "playful interaction" data. Fig. 1 and Fig. 2 display a few examples of playful interaction trajectories. We discuss data collection in more detail in Appendix A.

Learning Visual Representations from Play: In our work, we aim to show that pretraining models with playful interaction data is effective for downstream robotics tasks. We choose to use a BYOL [24] style framework to pretrain and learn a visual representation. Unlike the instance-based method used in BYOL, we explore a time-based [25, 26] approach to leverage the temporal association available in videos. Instead of augmenting a copy of the same frame, we augment a frame a few timesteps away in the same trajectory. Unlike [25], however, we do not require paired viewpoints of the same observation. We learn a representation purely from comparing observations from a single viewpoint at different timesteps. We find that a time-based approach is much more effective than the purely instance-based method used in BYOL. The full setup is explained in Appendix B.

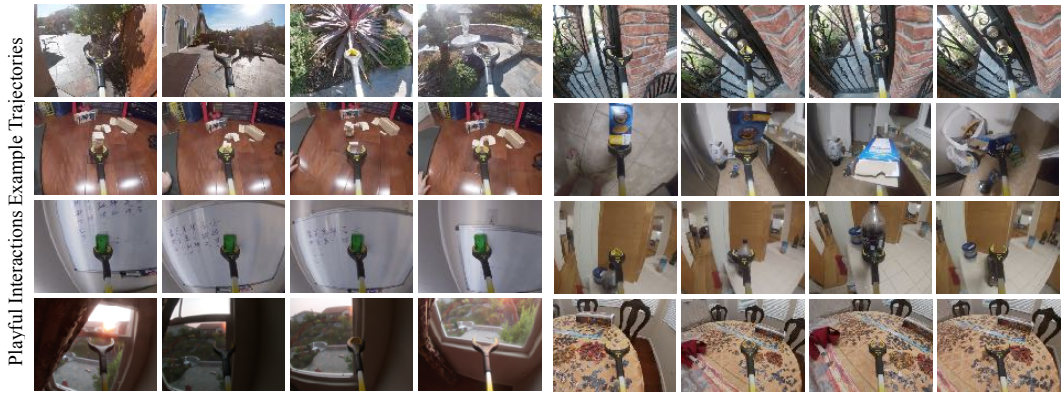

Figure 2: Some play examples are more free-form and undirected, such as walking around in an open space. Others contain repeated actions and suboptimal behaviors, such as dropping a bottle and knocking it over. Most importantly, these play trajectories are collected without specific instructions, making it diverse and easy to obtain.

Downstream Learning: After training on playful interaction data to learn a meaningful representation, we use this representation for downstream manipulation tasks. Unlike other works that utilize self-supervised contrastive pretraining [27, 28, 29, 24], our network architecture builds on top of the pretraining encoder and continues to update representation weights. More details in Appendix B.

Behavior Cloning: We learn a policy using behavior cloning [30, 31]. Each dataset contains observation-action pairs $D = \{(I_t, a_t)\}$, where I_t is an image and a_t is the action to get from I_t to I_{t+1}. Our task-training model takes in an observation image $I_t \in \mathbb{R}^{3\times 224\times 224}$ and learns a function $f(I_t, a_t)$ that maps observations I_t to actions a_t. Action labels are the relative changes in pose across frames and are provided by the dataset. Our objective is to minimize a mean squared error (MSE) loss computed on the predicted translation vectors.

3 Experiments

We designed our experiments to address four key questions. First, does self-supervised pretraining with playful interactions capture a diverse set of environments to improve visual imitation? Second, are play representations better than representations learned on ImageNet? Third, would task-specific representations do as well as task-agnostic representations learned from play? Finally, can play representations be combined with other modes of pretraining to get better performance? These final two questions are addressed in Appendix E, and to further understand the effects of play data and subsequent representation learning, we run a suite of ablations described in detail in Appendix F.

We evaluate our approach on two tasks, pushing and stacking. We use subsets of the 1000 pushing and stacking examples provided in [19]. The goal of the pushing task is to slide an object across a flat surface onto a red circle. The diverse dataset includes demonstrations of around 20 different objects in many diverse scenes, which makes accurately manipulating objects especially challenging. The prehensile stacking task requires grasping an object and placing it onto another object. We evaluate our method using MSE on 100 held-out video demonstrations in unseen environments for both tasks. Examples of pushing and stacking tasks are illustrated in Fig. 1.

Baselines: BC is a a behavior cloning policy trained from scratch, with data augmentations. AE and VAE are baselines pretrained on playful interaction data via an Autoencoder/VAE rather than BYOL. PLAY baselines are the models we train with playful interaction data. BC-OTHER is an ImageNet initialized baseline pretrained on other tasks, to really see how effective playful interaction data is for visual representation learning.

All baselines appended with *-I* are ImageNet pretrained baselines, where we load first a model with weights that have been trained for ImageNet classification tasks, rather than from scratch. We then use this pretrained model to run BC for our downstream manipulation tasks.

Table 1: Test Mean Squared Error of Playing and Stacking Task (lower is better).

Task	BC	AE	VAE	PLAY	BC-I	AE-I	VAE-I	PLAY-I	BC-OTHER
Push	0.095	0.101	0.084	**0.068**	0.08	0.093	0.085	**0.059**	0.085
Stack	0.137	0.139	0.135	**0.129**	0.126	0.138	0.137	**0.104**	0.128

Does Training on Playful Interactions Lead to Good Representations? To test whether self-supervised pretraining with playful interactions can learn a meaningful representation, we first train a model using our collected playful interaction data via BYOL. Then, we load the learned weights into our model to train on the downstream task. We train on 100 trajectories for both the pushing and stacking task. If our playful interactions can learn effective visual representations, we expect that this policy will outperform one where the downstream task is directly trained with BC from scratch. As shown in the first (BC) and fourth (PLAY) columns of Table 1, we see that our play model improve MSE from 0.095 to 0.068 in the pushing task and 0.137 to 0.129 in the stacking task. The performance gap is apparent when we visually compare actions, shown in Fig. 4 in the Appendix.

How does Pretraining on Playful Interactions Compare to ImageNet Pretraining? We study whether playful interactions are able to provide enough diversity and information in its learned representation to surpass the performance of BC-I (BC with ImageNet pretraining). The BC-I baseline is trained on significantly more data, but does not leverage playful interaction supervision, so we hypothesize that the baseline likely learns a representation better suited for more vision-based tasks. We first train a randomly initialized model to learn a representation from playful interaction data (PLAY). Using this model, we then learn a BC policy on the pushing and stacking task. The baseline BC-I is trained directly on the tasks with ImageNet-pretrained weights. Our results are shown in the fourth (PLAY) and fifth (BC-I) columns of Table 1. We find that our method performs better than ImageNet training for both the pushing and stacking stack. Qualitative results show predicted actions on held-out test data in Fig. 4 in the Appendix. Comparing BC-I and PLAY, we see there is an MSE of 0.08 to 0.068 in pushing and 0.126 to 0.129 in stacking. We further compare our method to the BC baseline trained on twice the number of demonstrations to evaluate whether our playful interactions can reduce the number of demonstrations needed to achieve good performance, which we discuss in Appendix F.1.

3.1 Connecting to Real Robot Results

The experimental results in this work are limited to offline MSE evaluations. However, to highlight our MSE evaluation results and contextualize it with real-robot evaluations, we can roughly base our results on Young et al. [19]. They show that a MSE of 0.028 corresponds to a 87.5% success rate for the pushing task and an MSE of 0.06 corresponds to 62.5% success rate for the stacking task on the real robot. Experimental results in our work have higher MSE since we are operating in the few-shot setting and hence use only a tenth of the pushing and stacking training data used in [19]. In the context of the experiments in this work, the BC baselines achieve a MSE of 0.08 and 0.126 for the pushing and stacking task respectively (Table 1), which both correspond to not being able to complete either task. The best performing models trained with our method achieve a MSE of 0.059 and 0.104 for the pushing and stacking task respectively, which roughly correlate to successfully solving the task around 60% for pushing and 29% for stacking.

4 Conclusion

We have presented an approach for learning downstream manipulation tasks via self-supervised pretraining on easy-to-obtain playful interaction data. Our method improves the generalizability of imitation learning baselines beyond simple data augmentations and provides significant improvements to current baselines. We demonstrate that our pretraining method can achieve comparable results to behavior cloning baselines using just half of the labeled task data. The success of our technique on simple behavior cloning opens up many exciting avenues for further work to incorporate play into more complex algorithms.

References

[1] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *ICRA*, 2018.

[2] B. C. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. *ICLR*, 2017.

[3] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098, 2017.

[4] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. *arXiv preprint arXiv:1802.09564*, 2018.

[5] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[6] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[7] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[11] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[12] P. Sharma, L. Mohan, L. Pinto, and A. Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *arXiv preprint arXiv:1810.07121*, 2018.

[13] J. Piaget. *Play, dreams and imitation in childhood*, volume 25. Routledge, 2013.

[14] C. Cook, N. D. Goodman, and L. E. Schulz. Where science starts: Spontaneous experiments in preschoolers' exploratory play. *Cognition*, 120(3):341–349, 2011.

[15] Z. L. Sim and F. Xu. Learning higher-order generalizations through free play: Evidence from 2-and 3-year-old children. *Developmental psychology*, 53(4):642, 2017.

[16] A. S. Lillard, M. D. Lerner, E. J. Hopkins, R. A. Dore, E. D. Smith, and C. M. Palmquist. The impact of pretend play on children's development: A review of the evidence. *Psychological bulletin*, 139(1):1, 2013.

[17] D. Whitebread, D. Neale, H. Jensen, C. Liu, S. L. Solis, E. Hopkins, K. Hirsh-Pasek, and J. Zosh. *The role of play in children's development: a review of the evidence*. 2017.

[18] S. Song, A. Zeng, J. Lee, and T. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3): 4978–4985, 2020.

[19] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto. Visual imitation made easy. *arXiv e-prints*, pages arXiv–2008, 2020.

[20] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

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

[22] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2169–2176. IEEE, 2017.

[23] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132, 2020.

[24] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[25] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. *Proceedings of International Conference in Robotics and Automation (ICRA)*.

[26] S. Purushwalkam and A. Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020.

[27] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[28] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[29] A. Stooke, K. Lee, P. Abbeel, and M. Laskin. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

[30] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[31] S. Ross, G. Gordon, and D. 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*, pages 627–635, 2011.

A Playful Interaction Data Collection Details

Each data collector used the same setup consisting of a reacher grabber stick and a GoPro camera and mount. Our guidelines are flexible enough that data collected from users are quite diverse. For example, demonstrations range from just a few seconds long to up to 18 minutes. Shorter demonstrations tend to be more task-based, while longer demonstrations typically involve many repeated movements and include more undirected interactions such as walking across a room. In total, we have 110 minutes and around 30,000 frames of playful interaction data. Our collected playful interaction data will be publicly available.

B Encoder Architecture

The setup of our approach consists of a self-supervised pretraining phase and a downstream imitation learning phase.

Following a BYOL approach, we train visual encoders $q(\cdot)$ and $k(\cdot)$ for the query and keys respectively and use a momentum-based update for the query encoder, similar to BYOL. The query encoder $q(\cdot)$ and key encoder $k(\cdot)$ each take in a single image $I_t \in \mathbb{R}^{3\times 224\times 224}$ and output a vector v. I_t, which is an augmented version of the frame at timestep t, is fed into the query encoder, and I_{t+3}, which is an augmented version of the frame at timestep $t+3$, is fed into the key encoder.

The play encoder architecture is as follows. Let Ck denote convolutional layers with k filters and Fk denote fully connected layers of size k. The base encoder architecture we use for play pretraining is simply the first three convolutional layers of the AlexNet: $C64 - C192 - C384$, followed by a pooling layer and a MLP projection head of size $F384 - F128$. We find that pretraining only the first three convolutional layers rather than four or five layers improves the model's ability to learn and generalize during downstream task evaluation and is key to good performance. In Appendix F.2, we provide analysis of pretraining at different layers.

The network architecture for downstream task learning consists of the base encoder used during pretraining followed by two additional convolutional layers and one projection layer. During training, weights from every layer are updated during task learning.

In Fig. 3, we show how we train via a time-based BYOL method with playful interactions. The intermediate layer highlighted in purple is the representation learned during pretraining, and the following two convolutional layers are trained only during task learning. The last layer is a projection to the predicted action.

C Training Details

We pretrain with 2 hours of playful interaction data, which is around 30,000 frames. The encoders are pretrained for 4,500 gradient steps with a batch size of 64. Then, the weights are loaded into the model for training downstream tasks, and trained for around 1,000 gradient steps. In both phases, we use a batch size of 64. During downstream tasks, we train on roughly 1400 images.

Our reported MSEs are the minimum error on our test dataset using the model with the best validation loss, which we found by performing a small scale hyperparameter search. One of the parameters in this search was the number of timesteps away frames should be in the self supervised pretraining. We found that using frames too near did not contain much temporal information, while frames too far apart often resulted in completely different scenes. Since supervised learning generally has little variance, we chose to just run 3 seeds of the BC-I baseline experiments, which had a standard deviation of 0.0009.

D Autoencoder and VAE Baselines

We find that generative methods such as AE and VAE do not work well. These baselines, which are first pretrained with playful interaction data and then run on downstream tasks, perform similarly to the BC baseline. We find that both AE and VAE are unable to reach the same level of accuracy as our method during training, and thus does not perform as well on held-out data during test time. The nature of the playful interaction data does not require a distribution to model actions, since there are

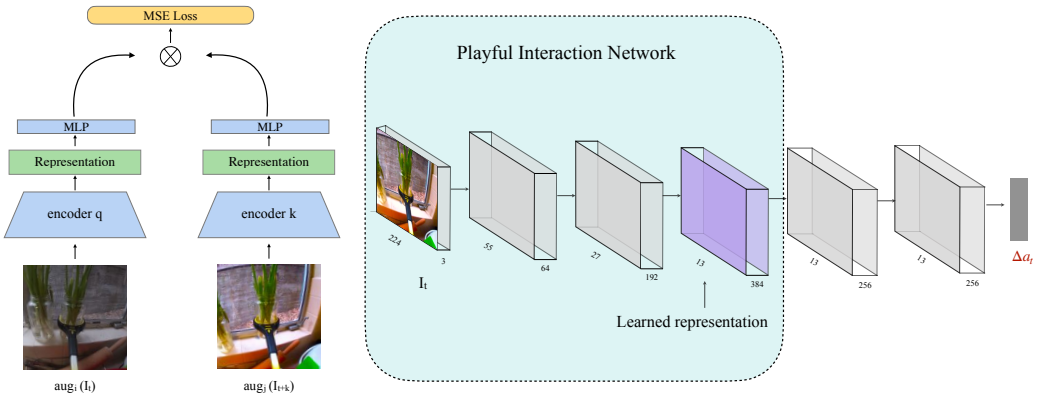

Figure 3: The setup of our approach consists of a self-supervised pretraining phase and a downstream imitation learning phase. (a) shows the first phase, where a representation is learned via time-based self-supervised learning. Each pair of images is fed through an encoder and projected to an MLP layer before computing similarity. (b) illustrates the encoder architecture for both phases. The purple layer is the representation we optimize during pretraining. The gray region is the encoder architecture used during playful interaction pretraining. The second phase of training (downstream task learning) updates all layers shown and outputs a predicted action. We leave out the projection MLP layers in this image for simplicity.

no decisions or points where there are diverging paths, and does not use temporal information, which could be why these models are not as effective as temporal-based momentum encoders. We also find that learning with instance discrimination does not improve performance and performs similarly to the BC baseline. This could be due to the fact that BC itself includes data augmentations.

E More Experimental Results

Does Pretraining on Other Task Specific Data Perform Similarly to Pretraining on Play? We further investigate whether the exploratory task-agnostic nature of playful interaction data is crucial to the learned representation, or if pretraining on another task in the same action space is able to learn a similarly effective representation. To this end, we compare a model pretrained on playful interaction data (PLAY-I) and a model pretrained on a different task (BC-OTHER). Specifically, we test whether a playful interaction-pretrained model outperforms a stack-pretrained model when trained on the pushing downstream task, and vice versa for the stacking task. We note that stacking and pushing have some structural similarity in actions, and that may improve those results. However, we find that when learning the pushing task, pretraining on the stacking data leads to no visible improvement in our experiments. We hypothesize that the pretraining phase is overfitting to the stacking data, and thus does not learn a generalizable representation. We see similar results for the stacking task in the last two columns of Table 1, where pretraining on a different task-specific dataset does not help the model learn a good visual representation for training other downstream tasks. We also experimented with pretraining on a completely different dataset, such as MIME, but found that the representations learned are not effective, likely due to the difference in setup and viewpoint. This further shows the effectiveness and importance of using playful interactions to learn a representation that can be used to efficiently learn downstream tasks.

Can Play Pretraining be Combined with ImageNet Pretraining to Learn Better Representations? We also evaluate our method combined with state-of-the-art pretrained baselines. In this set of experiments, we demonstrate that by combining our play-pretrained model with ImageNet pretraining (PLAY-I), we can achieve even better performance: 0.059 and 0.104 for pushing and stacking respectively. We first initialize our model with ImageNet weights before pretraining on playful interaction data. We then train using BC on downstream tasks. As shown in the fifth and second to last columns of Table 1, PLAY-I performs much better than the BC-I baseline, a 26% improvement for pushing and 19% improvement for stacking. Furthermore, PLAY-I, combined with ImageNet pretraining, outperforms PLAY (fourth column).

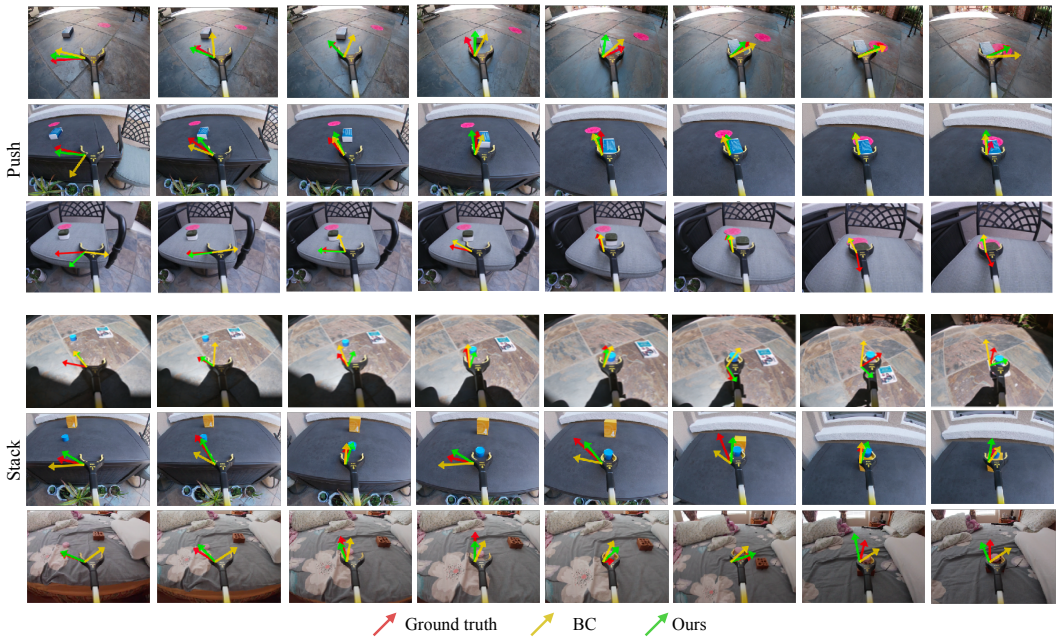

Figure 4: Here we present qualitative results of our experiments for both tasks. Each overlaid arrow on the images represents the action (transverse plan only) predicted by each policy.

F Additional Ablation Studies

F.1 Comparison with More Downstream Task Data

For both the pushing and stacking downstream task, we evaluate our method on 100 trajectories. We have shown that our playful interaction pretraining method is able to significantly improve upon baselines using the same number of expert pushing and stacking demonstrations. However, we also want to evaluate how our method performs compared to using more expert data. To this end, we test the baseline BC-I model on 200 trajectories and compare that to our playful interaction model (PLAY) trained on only 100 trajectories.

Table 2: Comparison of Amount of Expert Demonstration Data

Task	BC-I		PLAY-I
# Expert demonstrations	100	200	100
Push	0.08	0.069	0.059
Stack	0.126	0.099	0.104

The pushing task trained with PLAY achieves an MSE of 0.059, which is 14% better than BC with 200 demonstrations despite training on only half the number of labeled data. In the stacking task, our method (0.104) is able to surpass BC performance (0.126) and nearly match BC trained on 200 demonstrations (0.099). We note that in the stacking task, although our method is only able to nearly match performance of using twice the amount of data, it is still a significant improvement upon the baseline. Results are shown in Table 2.

F.2 Effect of Pretraining at Earlier Layers

We experiment with representation learning at earlier layers of the base encoder. Specifically, we study the effects of pretraining a representation at the third, fourth, and last convolutional layers shown in Fig. 3. We find that representation learning at earlier layers and updating all weights during task learning is crucial to learning a good policy. When learning a representation at the final

convolutional layer and fine-tuning only on the last linear layer(s), the policy is unable to learn enough during fine-tuning to successfully complete the task, and performs worse than BC-I (0.093). This suggests that the task-agnostic learned representation itself still needs signal from task-training. If we update all the weights during task learning, however, the model is unable to generalize well and achieves performance (0.081) similar to that of BC-I. Thus, learning a representation at an earlier convolutional layer and updating all weights enables sufficient learning during imitation while preserving the ability to generalize to test time scenarios. Fig. 5 shows the ablation study comparing performance at different layers of pretraining for both downstream tasks.

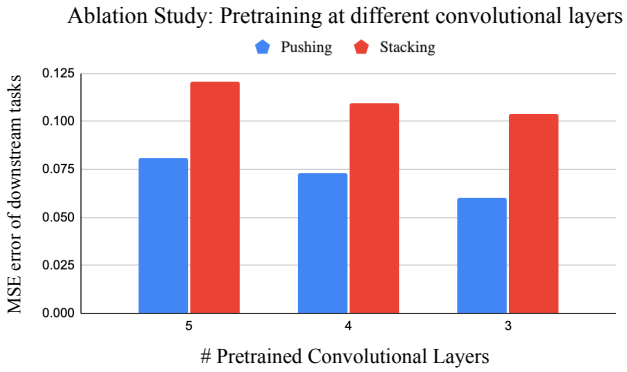

Figure 5: Learning comparisons between pretraining until different layers. We perform ablations over pretraining with playful interaction data until the third, fourth, and fifth convolutional layer of the model. Note that during downstream task training, we train on the full architecture regardless of which layer we pretrained until. The y-axis is MSE error of predicted and ground truth actions in log scale. We find that pretraining fewer layers significantly improves downstream task performance. We hypothesize that this happens because during pretraining, since we are not optimizing for the same objective as during downstream task learning, the model tends to overfit when trained on more layers.

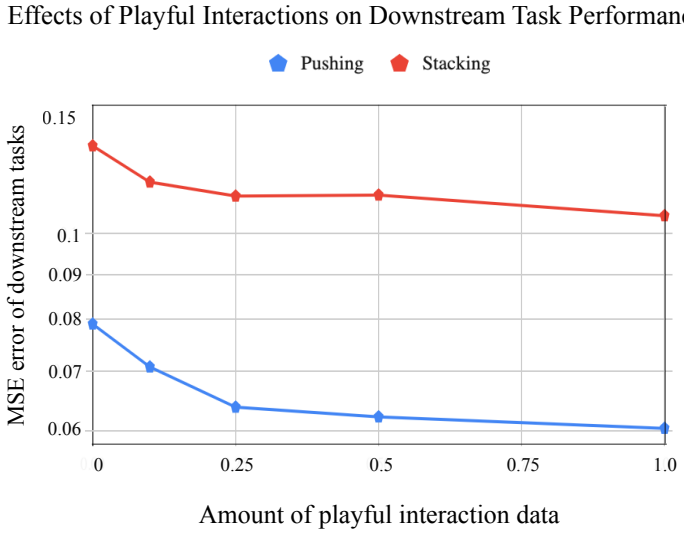

Figure 6: Learning comparisons between different amounts of play data. Note that the y-axis is MSE error of predicted and ground truth actions in log scale. We see that more play data greatly improves performance on downstream tasks, and diminishing improvements suggest that large-scale play data is needed for improved accuracy.

F.3 Amount of Play Data

We study the effects of the amount of play data on downstream task performance. We have 110 minutes of playful interaction videos split into around 30,000 frames and evaluate how important this data is to task learning. Fig. 6 shows the performance of the pushing and stacking task when pretrained on the following fractions of play data: 1, 10, 25, 50, and 100%. We note that as the amount of playful interaction data we use increases, the MSE error decreases significantly, especially when the amount of play data is low. Training with just 25% of our play data shows a 13% increase in performance when compared to training with 1% of play data. However, in both the pushing and stacking task, the improvements start to diminish as we approach two hours of data, suggesting that large-scale playful interaction data is needed for further performance gains.