World Models for Multi-task Robotic Pretraining

Jared Mejia
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
jamejia@cs.cmu.edu

Mohan Kumar
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
mohankus@cs.cmu.edu

1 Introduction

Robotic manipulation has made significant progress in recent years with various methods showcasing the capability of current systems to achieve success on specialized tasks provided there is adequate data [4, 13, 1, 2, 53]. However, the current paradigm requires collecting entirely new data and training another policy from scratch for each new task, making it difficult to develop fully autonomous agents capable of performing a wide range of tasks in any real-world setting.

Ideally, robotic agents would instead exhibit positive transfer [5], whereby prior experiences and transferable skills can facilitate learning and performance on new tasks. Humans are known to benefit most from positive transfer in scenarios where there is a certain degree of similarity or overlap between two tasks or domains, allowing sharing of knowledge between tasks and enhanced performance in novel situations [22, 42, 8]. Various methods in the fields of natural language processing [39, 46, 36] and computer vision have already benefited greatly from the shared structure between related tasks in the respective fields, allowing them to leverage massive amounts of data not entirely specific to the task of interest. While the tasks of these fields may be more analogous to the high-level cognitive tasks from psychology literature, we would expect there to be similar if not greater potential for positive transfer between low-level control tasks in manipulation, as all such tasks adhere to the natural laws of physics.

Multitask learning approaches that leverage diverse data sources have been explored in various ways. Some researchers have focused on optimizing the data collection process, aiming to efficiently gather large-scale data and using simple behavior cloning policies for downstream tasks (e.g., [49, 3, 7]. Although performance in robotic manipulation tasks seems to scale with the amount of data, it is likely that additional methods could be combined with this approach to further benefit from the collected datasets.

Another recent trend in multitask robotic learning has focused on obtaining a universal pretrained visual representation (PVR) [35, 31], with the goal of using a single visual encoder for any downstream robot learning task. However, current PVRs do not always improve performance, and recent work has shown that training from scratch often outperforms fixed pretrained visual backbones [20]. Furthermore, current approaches to PVRs typically use egocentric videos of humans acting in the real world as their primary source of data [11, 35, 31]. Consequently, there is a large domain shift when these encoders are applied to robotic tasks, and the representations may not necessarily encode reasoning about the consequences of robot actions. To address this problem, recent work (PTR) has used end-to-end offline RL for both pre-training and fine-tuning. Nevertheless, offline RL methods are known to be unstable to train and highly sensitive to hyperparameters, particularly when using images as observations.

In this work, we aim to devise a method for learning to multitask representations while addressing both the issue of domain shift, as is the main problem with PVRs, and training instability, the main shortcoming of end-to-end policies. Our key insight is that world models [15, 16] are a mechanism with the potential for learning generalizable multi-task representations that reason about

10707 Class Project (2023).

the consequences of an agent's actions. While world models have for the most part only been applied in the online learning setting [15, 18], we demonstrate their applicability in the offline setting as well. Our method allows us to learn a policy with either imitation learning or offline RL methods using fixed representations obtained from our world model for image inputs and conditioning on task identifiers. At test time, we combine our frozen pretrained world model with the learned policy and perform rollouts by updating the belief and state of world model at each timestep and passing it as input to the policy. Our results demonstrate that multi-task world model pretraining achieves positive transfer, improving the overall success rate across all tasks in comparison with single-task policies, and outperforming equivalent end-to-end multi-task policies.

2 Related Work and Background

Representation Learning in Robotics A variety of approaches have been taken to address the problem of general representation learning from images for robotic manipulation. The two main paradigms consist of learning representations from entirely in-domain data or making use of large amounts of unrelated task data. The methods that fall into the first category generally make use of data augmentation or contrastive learning [28, 26, 37, 40]. Prior work has already utilized the learning of latent space models for representation learning [10, 17, 15, 12], however, most of these approaches have only been applied in the online model-based RL setting. To our knowledge, our method is the first attempt to apply learned dynamics models for multi-task representation learning in the offline setting.

The second class of representation learning methods has focused mostly on scaling the pretraining to large datasets. The idea is that with enough diversity and scale of the data, the domain gap will be diminished and a universal representation will emerge. [48, 45, 23, 44]. More recently, several works have aimed to make use of egocentric videos of humans acting in diverse environments, in an attempt to lessen the domain shift between pretraining data and robotic data [35, 30, 31]. Despite these efforts there remains no universal pretrained representation that outperforms all others on any given downstream task, and recent work has shown that learning a robot policy entirely from scratch often outperforms learning from fixed pretrained representations [20].

World Models World models [12, 15, 16, 18, 19] have proven to be an effective approach to data-efficient reinforcement learning in simulation and, more recently, in real-world robotic environments for locomotion and manipulation [47]. With only a small amount of real-world interaction, learned world models enable planning and behavior learning by predicting the future states and rewards that would result from taking certain actions [6, 47]. As world models implicitly encapsulate knowledge about the physical dynamics of an environment, they align well with the motivation of making use of large amounts of multi-task data and prior works have explored learning a dynamics model in a self-supervised manner through environment interactions that generalizes to diverse downstream tasks and objects [6, 43, 34, 33]. Furthermore, recent work has applied this notion of world modesls to static datasets in the offline RL setting [50, 41]. Whereas current model-based offline RL approaches [50, 41, 52, 24] learn individual dynamics models for each task, we aim to learn a shared dynamics model across all tasks in our experiments. World models have been applied to the offline RL setting (COMBO, LOMPO), however, they struggle with the problem of overestimating the values of state-action pairs—a fundamental problem of offline RL that is exacerbated when the replay buffer is mixed with imagined rollout transitions from the learned model. Our method aims to circumvent this issue by only using the learned world model for encoding our offline data, rather than generating additional data. The problem of estimating values of state-action pairs in the support of the offline data is offloaded to IQL [25], a method which never needs to evaluate actions outside of the dataset by treating the state value function as a random variable with randomness determined by action and taking a state conditional expectile of the random variable to estimate the value of a state.

3 Methods

3.1 Approach

Our approach to learning from pixels in multitask robotic settings involves decoupling representation learning from policy learning. In particular, we learn a shared world model across all tasks in our

offline data. We then freeze our world model and save the latent state representation generated by our world model based on the pixel observations in our dataset. We use these latent representations to compose our observations and learn an MLP policy from the embeddings. During inference time, we use the world model to encode the incoming image and update the belief, and then pass the latent representation to the policy.

3.1.1 Multi-task offline World Model pre-training

We first learn a single World Model for all tasks $i \in \mathcal{T}_{\text{train}}$ with the Recurrent State-Space Model (RSSM) [[15]], using CNNs [[29]] for the image encoder and decoder:

Image encoder:	$h_t = E_\theta(o_t)$
Inference model:	$s_t \sim q_\theta(s_t \vert h_t, s_{t-1}, a_{t-1})$
Latent transition model:	$s_t \sim \widehat{T}_\theta(s_t \vert s_{t-1}, a_{t-1})$
Reward predictor:	$r_t \sim p_\theta(r_t \vert s_t)$
Image decoder:	$o_t \sim D_\theta(o_t \vert s_t)$

We follow the latent space representation from [15] $s_t = [d_t, z_t]$ consisting of deterministic d_t and a sampled stochastic representation z_t :

Deterministic State Model:	$d_t = f_\theta(d_{t-1}, z_{t-1}, a_{t-1})$
Stochastic Inference Model:	$z_t \sim q_\theta(z_t \vert h_t, d_t)$

We follow [15] in optimizing the components jointly by maximizing the variational lower bound which can be shown to be decomposed into reconstruction terms for the observations and rewards and a KL regularizer:

$$\mathcal{L}_{REC} = \mathbb{E}_P \left[\sum_t \left(\mathcal{L}_O^t + \mathcal{L}_R^t + \mathcal{L}_D^t \right) \right] + c$$

$$\mathcal{L}_R^t = \ln q_\theta(r_t \vert s_t) \qquad \mathcal{L}_O^t = \ln D_\theta(o_t \vert s_t)$$

$$\mathcal{L}_D^t = -\beta \text{KL}\left(p_\theta(s_t \vert s_{t-1}, a_{t-1}, o_t) \vert\vert q_\theta(s_t \vert s_{t-1}, a_{t-1}) \right)$$

We also experimented with a variant of the state encoder and decoder using both images and the robot's proprioceptive state. In this variant, we encode the image with the same image encoder E_θ, concatenate the proprioceptive state with the encoded image, and pass this through a 2-layer MLP, using the resultant vector as the new h_t. Similarly, we modified the decoder to reconstruct both the proprioceptive state and the pixels of the original input image. A priori, the idea that passing more state information to the model would result in a better representation seemed reasonable. However, we found that the additional loss of reconstructing the proprioceptive state appeared to be too strong of a signal, leading to the latent model effectively overfitting to the proprioceptive state as evidenced quantitatively by the increased model reward loss as well as qualitatively in the resultant generated imagined trajectories. These results suggest that for manipulation tasks, learning the dynamics directly from pixels alone yields a more robust and generalizable representation and dynamics model. For this reason, we chose not to condition any of the world model components on the task identifier either.

Implementation details We build on an implementation of Dreamer [15] from TorchRL [38] to learn our world model. We train a single online SAC [14] agent from the MT10 Metaworld benchmark [51] garage implementation [9] and perform rollouts in the MT10 environments, collecting a total of 130 successful trajectories from 6 different environments with 6-DOF action space. We provide more details on the environments in the Results section.

While training the world model, we sample a batch of 256 sub-trajectories at random, where each sub-trajectory has length 25 environment steps. We use a state dimension of size 256 for d_t, and a belief dimension of size 256 for z_t, resulting in a latent state of size 512. We train and save the model after 3,000 optimization iterations, taking about 90 minutes on a NVIDIA GeForce RTX 3080 Ti. We then reiterate through all of the trajectories, pass them through the world model, and save the resultant latent space representation $s_t = [d_t, z_t]$ for each timestep t.

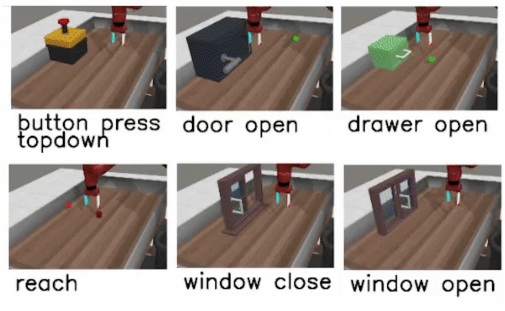

Figure 1: Our 6 evaluation tasks from the Metaworld benchmark [51]

3.1.2 Multitask offline policy learning

Once we have the latent state representation for all timesteps for our multi-task data, we concatenate this state with a one-hot encoded task-identifier as well as a 4-dimensional vector consisting of the Cartesian end effector coordinates and a measurement of how open the gripper is. We learn both a multitask offline RL policy and a multitask behavior cloning policy using the concatenated vectors as observations.

Multitask offline RL policy We use IQL [25] as the method for learning our multitask offline RL policy. We build on an implementation of IQL from TorchRL [38]. The actor, critic, and value networks are all 2 layer MLPs. The actor parameterizes a multivariate Gaussian distribution with a diagonal covariance matrix and dimension equal to the action dimension. We use dropout (cite dropout) equal to 0.1 for the actor network. We use an inverse temperature $\beta = 0.5$ and an expectile $\tau = 0.7$. We sample batches of size 256 and train for 300,000 optimization steps which takes approximately 80 minutes on a NVIDIA GeForce RTX 3080 Ti.

Multitask BC policy Our BC policy is a 2-layer MLP similar to the IQL networks except its output dimension is equal to the action dimension and it is trained to regress on the actions from the demonstration data with a mean-squared error MSE objective. The BC network has a dropout value of 0.1. We use only 150,000 optimization steps to prevent overfitting which takes just 10 minutes on a NVIDIA GeForce RTX 3080 Ti.

3.2 Baselines

IQL End-to-End We train one IQL agent end-to-end (E2E) for each individual environment and a single IQL E2E agent across all tasks. For all IQL E2E agents, we encode visual inputs in a similar manner to PTR [27]. In particular, we use a CNN with group normalization layers [47, 21], and learned spatial embeddings [27], the latter meant to help stabilize the training of Q-functions from images in offline RL. We then concatenate the encoded image with the state vector and pass the concatenated input to an MLP. The state vector consists of the 4-dimensional state, as well as the one-hot task identifier for the multitask agent.

BC End-to-End We train one BC E2E agent for each individual environment and a single BC E2E agent across all tasks. We use the same architecture as the IQL E2E actor-network, except the BC agents are trained to regress on actions from the demonstration data with an MSE objective.

4 Experiments and Results

The goal of our experiments is to determine if: **(a)** positive transfer can be achieved for policies by training on multiple tasks and **(b)** using World Models for learning visual representations from multi-task data leads to better policies in comparison with training end-to-end or using static image representations such as R3M.

4.1 Offline Data

Our offline dataset consists of successful demonstrations from 130 rollouts in 6 different MT10 Metaworld environments, totaling 780 trajectories. Each of the environments consists of a Sawywer Robot Arm with a different task specification for each environment. Each environment has 50 variations of the starting state of the environment and the goal location or object. We train an MTSAC agent from the original Metaworld paper [51] as implemented in [9] on 10 variations of the environment. We then performed rollouts with the MTSAC agent on the 10 variations of the environment and collect 130 successful demonstrations total per environment.

4.2 Evaluation

For evaluation, we perform a single rollout of each trained agent on every variation of all 6 environments, totaling 300 rollouts per agent. Note that this includes at least 40 variations of every environment guaranteed to not be seen in the offline data, making the evaluation not only a test of the performance between tasks but also the performance with deviations from the training data within the same environments. We record statistics of the reward and success rates on each environment for every agent. The results are shown in Table 1

Table 1: Task performance for six different tasks using Multitask from Scratch, Individual experiment methods, and Multitask with world model. Mean and standard deviation are shown for each task and experiment method combination.

Task	**Multitask from scratch**		**Individual**		**Multitask World Model (Ours)**	
	IQL	**BC**	**IQL**	**BC**	**IQL**	**BC**
button-press-topdown-v2	0.04	0.12	0.14	**0.32**	0.2	0.3
door-open-v2	0.42	0.7	**1**	0.98	0.92	0.9
drawer-open-v2	0	0.02	0	0.38	0.74	**0.82**
reach-v2	0.04	0.24	**0.94**	0.26	0.22	0.2
window-close-v2	0.28	0.5	0.42	0.84	0.96	**1**
window-open-v2	0.08	0.4	0.72	0.64	**0.84**	0.72
Overall Success Rate	0.1433	0.33	0.5367	0.5733	0.6466	**0.6567**

4.3 Results

There are a few takeaways to be made from the results in Table 1. We first note that the world model for both IQL and BC only uses sparse 0/1 rewards during training, as does the multitask IQL that we use together with the world model. In contrast, IQL multitask from scratch and individual IQL uses dense rewards during training, which may be unavailable in real-world offline datasets.

We see that the performance of IQL uniformly increases between training multitask IQL from scratch and training multitask IQL using World Model pretraining and representations during rollouts. This suggests that in multitask settings, our world model pretraining and representations do lead to an increase in performance. Similar results are shown for multitask BC with the world model, where there is only a single task in which the multitask BC trained from scratch outperforms the multitask BC with the world model.

We see that across most tasks, the performance of the multitask model is competitive with the models trained on a single environment, suggesting that no significant negative transfer is occurring from our method in multitask training. Furthermore, the overall success rate between the multitask world model approaches significantly improves over the multitask from scratch approaches, and substantially improves over the specialized agents. Importantly, we note that shared world model representations leads to a significant increase in the drawer-open-v2 task, suggesting that for difficult tasks, using a shared learned dynamics model may allow for knowledge transfer from other tasks which can prove to be useful for the task at hand.

Finally, we note that there is little difference between the overall success rates between IQL and BC, despite the fact that we are learning from near optimal demonstrations and using multitask data. This is in contrast to results from prior works that either claim that BC significantly outperforms offline

RL with optimal demonstrations [32] or offline RL outperforms BC in multitask pretraining regimes [27]. Here, it appears that the factor that matters most is the method of representation learning.

5 Discussion

We present an approach to learning multitask representations using shared world models across tasks. Our experiments demonstrate strong evidence of positive benefits from learning a shared world model across tasks and decoupling representation learning from policy learning. We hope to perform more extensive comparisons between our approach and other methods of learning a pretrained visualization in future work. As our current experiments only consist of expert data, we additionally hope to extend our experiments to few-shot learning settings in which we only have access to a small amount of data for a target task. We also plan on testing our approach in settings for which we have access to a large amount of diverse suboptimal data, as intuitively we would expect a learned dynamics model to improve with more data, regardless of whether the data is optimal or not. Finally, we hope to demonstrate the effectiveness of our method in real-world robotic settings and settings where we do not have access to action annotations, such as human videos.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2022.

[3] 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.

[4] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand dexterous manipulation from depth.

[5] Daniel Ed Druckman and Robert A Bjork. *Learning, remembering, believing: Enhancing human performance.* National Academy Press, 1994.

[6] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[7] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[8] David Epstein. *Range: Why generalists triumph in a specialized world*. Penguin, 2021.

[9] The garage contributors. Garage: A toolkit for reproducible reinforcement learning research. `https://github.com/rlworkgroup/garage`, 2019.

[10] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[12] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[13] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022.

[14] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[15] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[16] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 09–15 Jun 2019.

[17] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[18] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[19] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[20] 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. *arXiv preprint arXiv:2212.05749*, 2022.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[22] Charles Hubbard Judd. *Psychology of high-school subjects*. Ginn, 1915.

[23] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022.

[24] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[25] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[26] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[27] Aviral Kumar, Anikait Singh, Frederik Ebert, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022.

[28] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[29] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[30] 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. *arXiv preprint arXiv:2210.00030*, 2022.

[31] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Dhruv Batra, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*.

[32] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[33] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Alan: Autonomously exploring robotic agents in the real world. *arXiv preprint arXiv:2302.06604*, 2023.

[34] Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391, 2021.

[35] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[36] Sharan Narang and Aakanksha Chowdhery. Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance, 2022.

[37] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[40] 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*, pages 416–426. PMLR, 2023.

[41] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pages 1154–1168. PMLR, 2021.

[42] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[43] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020.

[44] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

[45] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[47] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pages 2226–2240. PMLR, 2023.

[48] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

[49] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *Conference on Robot Learning*, pages 1992–2005. PMLR, 2021.

[50] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

[51] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[52] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[53] Kevin Zakka, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. RoboPianist: A Benchmark for High-Dimensional Robot Control, 2023.