GENERALIST WORLD MODEL PRE-TRAINING FOR EFFICIENT REINFORCEMENT LEARNING

Yi Zhao[1] **Aidan Scannell**[1,2] **Yuxin Hou**[3] **Tianyu Cui**[1,8] **Le Chen**[4]
Dieter Büchler[4,5,6] **Arno Solin**[1] **Juho Kannala**[1,7] **Joni Pajarinen**[1]

ABSTRACT

Sample-efficient robot learning is a longstanding goal in robotics. Inspired by the success of scaling in vision and language, the robotics community is now investigating large-scale offline datasets for robot learning. However, existing methods often require expert and/or reward-labeled task-specific data, which can be costly and limit their application in practice. In this paper, we consider a more realistic setting where the offline data consists of **reward-free** and **non-expert multi-embodiment** offline data. We show that generalist world model pre-training (WPT), together with retrieval-based experience rehearsal and execution guidance, enables efficient reinforcement learning (RL) and fast task adaptation with such non-curated data. In experiments over **72 visuomotor tasks**, spanning **6 different embodiments**, covering hard exploration, complex dynamics, and various visual properties, WPT achieves 35.65% and 35% higher aggregated scores compared to widely used learning-from-scratch baselines, respectively.

1 INTRODUCTION

With the success of scaling laws in vision and language (Brown et al., 2020; He et al., 2022; Kirillov et al., 2023; Touvron et al., 2023), the robotics community has started investigating the use of large-scale offline datasets for robot learning (Brohan et al., 2023b; Walke et al., 2023; Brohan et al., 2023a; O'Neill et al., 2023; Khazatsky et al., 2024; Team et al., 2024). Large-scale offline datasets enable the trained agents to learn a wide range of skills as well as help the agent adapt to new tasks with limited samples (Team et al., 2024).

To leverage offline datasets for decision-making, imitation learning (IL) (Brohan et al., 2023b; Team et al., 2024; Doshi et al., 2024) and offline RL (Levine et al., 2020; Kumar et al., 2020; Fujimoto & Gu, 2021; Kumar et al., 2023) are commonly used. However, these methods have limitations in terms of both data and algorithmic aspects. IL assumes the availability of expert data from human demonstration or specialist RL agents, which usually involves an expensive data collection procedure. Further to this, agents learned by IL are usually sensitive to environmental perturbations such as action delay, lighting conditions, sensor noise, etc (Chae et al., 2022; Zare et al., 2024). Offline RL aims to learn agents from offline data but suffers from training instability, especially for pixel-based observations (Kumar et al., 2020; Lu et al., 2023). Besides the training stability issue, offline RL requires task-specific datasets, labeled with rewards. This poses another challenge for many real-world applications, e.g., robotic manipulation, to utilize offline datasets for *new* tasks, it is necessary to retrospectively annotate the image-based offline data with rewards, which can be challenging and laborious. Furthermore, fine-tuning an offline RL agent, so-called offline-to-online RL (Nair et al., 2020; Lee et al., 2022; Zhao et al., 2022; Yu & Zhang, 2023; Nakamoto et al., 2024), requires additional tricks to stabilize training and prevent performance collapse caused by the distributional shift in the fine-tuning stage.

In contrast to IL and offline RL, which require curated datasets, many real-world scenarios have data from various sources. To this end, we focus on the more realistic setting where the offline dataset

[1]Aalto University, Finland [2]University of Edinburgh, UK [3]Deep Render, UK [4]Max Planck Institute for Intelligent Systems, Germany [5]University of Alberta, Canada [6]Alberta Machine Intelligence Institute (Amii) [7]University of Oulu, Finland [8]Imperial College London, UK.
Correspondence to: Yi Zhao yi.zhao@aalto.fi

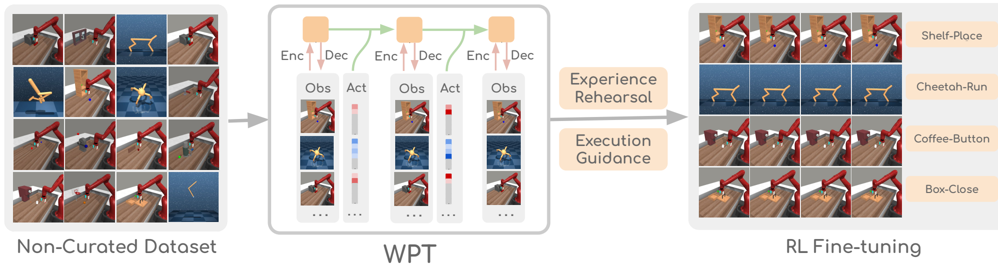

Figure 1: Overview of generalist world model pre-training (WPT). Facing non-curated offline datasets with reward-free, non-expert, and multi-embodiment data, we train a task and embodiment-agnostic world model. The pre-trained world model improves the sample efficiency of RL training over a wide range of tasks.

includes non-curated mixed-quality data consisting of *reward-free* and *non-expert* data, collected by multiple agents with varying embodiments. This setting has a minimal requirement for offline data, which significantly expands the pool of usable data. The primary research question we aim to answer is:

What is the best way to leverage non-curated offline data for efficient multi-task & multi-embodiment learning?

Definition 1.1 (Non-curated data)**.** Non-curated data encapsulates all observation-action trajectories, i.e., it can consist of reward-free and non-expert data for any embodiment.

Existing methods for leveraging offline data (e.g. IL and offline RL) typically fail given non-curated data. For example, IL would require manually filtering the dataset to select expert trajectories and offline RL would require retrospective reward labeling. With such non-curated data, a common strategy is to pre-train a visual encoder (Schwarzer et al., 2021; Nair et al., 2022; Parisi et al., 2022; Xiao et al., 2022; Yang & Nachum, 2021; Shang et al., 2024). However, visual pre-training alone fails to fully leverage the rich information that exists in the offline dataset, e.g., dynamics model, informative states, or action prior, etc. Table 1 compares approaches for leveraging offline data.

Facing the non-curated offline dataset, training a world model seems a “natural” choice, as it holds the promise of better generalization and better sample efficiency. However, model-based approaches are not commonly used in the offline-to-online setting (Rafailov et al., 2023; Wu et al., 2024). This is partially due to the performance margin between model-based and model-free approaches not being significant (Yu et al., 2020; Kidambi et al., 2020) while model-based approaches involve higher training complexity. In addition, in the typical offline-to-online RL setting, training an offline agent with model-based approaches can be unstable, especially for pixel inputs (Lu et al., 2023), and strong regularization is required during the online training stage (Rafailov et al., 2023).

In this paper, we show that with careful design choices, world model pre-training unlocks more usable offline data and better fits the pre-train and fine-tune paradigm used for scaling. Our key insights are *i)* instead of pre-training a policy using model-based offline RL as done in previous work (Yu et al., 2020; Kidambi et al., 2020; Rafailov et al., 2023), learning a world model is stable and scalable, *ii)* unlike previous methods (Hafner et al., 2020; 2021; 2023; Hansen et al., 2024) which learns a world model per task, by simply padding actions to unify the action spaces, a single world model can be pre-trained on the non-curated offline data consisting of data sources from different tasks and different embodiments, *iii)* via fine-tuning, a pre-trained generalist world model can boost RL's sample efficiency over a wide range of tasks and embodiments, *iv)* during fine-tuning, experience rehearsal, and execution guidance can largely improve the performance.

We extensively evaluate WPT on 72 pixel-based continuous control tasks covering locomotion and manipulation with different action spaces, hard exploration, high dimensions, and complex dynamics. Under a limited sample budget (150k samples), WPT outperforms previous SOTA methods by a decent margin. Specifically, WPT obtains 35.65% and 35% higher normalized scores compared to DrQ v2 and Dreamer v3 under the same sample budget and matches baselines' results obtained with higher sample budgets (500k samples for DMControl and 1M samples for Meta-World.). For example, WPT enables an agent learning to control an Ant robot to walk forward within 100 trails, while widely used learning-from-scratch baselines need **10-30**$\times$ samples. We further demonstrate that, without any modifications, WPT also helps task adaptation, where the agent is required to

Table 1: Comparison with different policy learning methods that leverage offline data.

METHODS	REWARD-FREE OFFLINE DATA	NON-EXPERT OFFLINE DATA	X-EMBODIMENT OFFLINE DATA	TRAINING STABILITY
OFFLINE RL		✓		
OFF2ON RL		✓		
RLPD		✓		
MT IL	✓		✓	✓
MT OFFLINE RL		✓	✓	
WPT (OURS)	✓	✓	✓	✓

continually adapt its skill to new tasks. Our results demonstrate strong performance with a large margin over model-free methods when leveraging *non-curated* offline data. We hope that our results motivate further investigation of model-based approaches for using offline data. In summary, our contributions include:

- We propose a new and more realistic setting for leveraging offline data where the offline data consists only of reward-free and non-expert multi-embodiment data.
- We show that generalist world model pre-training along with a series of careful design choices achieves strong performance in 72 visuomotor tasks spanning 6 embodiments.
- To facilitate further research with non-curated offline data, we open-source our datasets, including 60K trajectories and 10M transitions from the DMControl and Meta-World benchmarks.

2 MAIN RESULTS

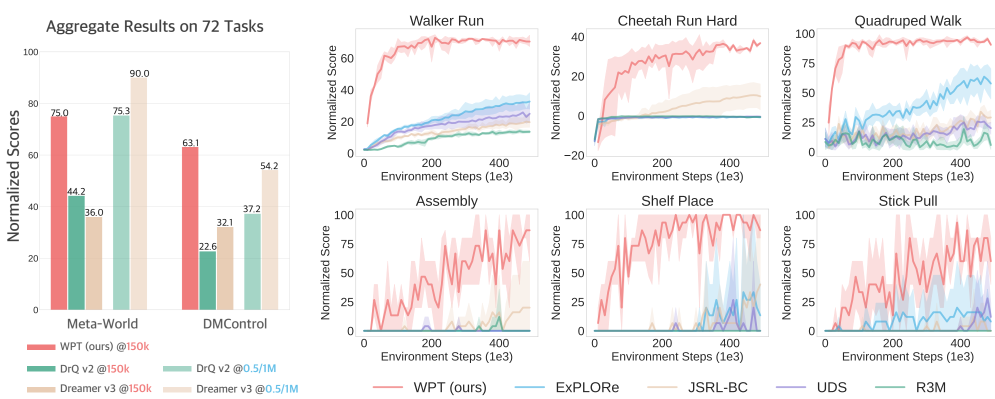

Figure 2: **Left:** Aggregate performance on **50 manipulation tasks** from Meta-World and **22 locomotion tasks** from DMControl with pixel inputs. We pre-train one *single* world model per benchmark on *non-curated* offline data, which can be fine-tuned to largely improve RL's sample efficiency. With 150k online samples, WPT obtains 35.65% and 35% higher normalized scores compared to DrQ v2 and Dreamer v3 using the same samples and matches baselines' results obtained with higher sample budgets (500k samples for DMControl and 1M samples for Meta-World). **Right:** WPT can effectively leverage non-curated offline data compared to baselines. We include 6 tasks from DMControl and Meta-World with pixel inputs. The mean and the corresponding 95% confidence interval across 3 seeds are plotted.

In this section, we present our main results to show that WPT improves the sample efficiency of RL training. We further include task adaptation results and ablation study in App. B

Dataset Our dataset consists of data from two different benchmarks: DeepMind Control Suite (DMControl) and Meta-World, visualized in Fig. 1 and App. F for the full list. For DMControl, we include 10k trajectories covering 5 embodiments collected by *unsupervised RL agents* (Pathak et al., 2017; Rajeswar et al., 2023), which are trained via curiosity without any task-related information. For Meta-World, we collect 50k trajectories spinning 50 tasks by executing the pre-trained RL agents from TDMPCv2 (Hansen et al., 2024) with injected Gaussian noise with $\sigma = [0.1, 0.3, 0.5, 1.0, 2.0]$ to mimic imperfect demonstrations. In total, the offline dataset consists of 60k trajectories with 10M state-action pairs from 6 different embodiments.

Tasks We evaluate our method on six selected *pixel*-based continuous control tasks from DMControl and Meta-World. The chosen tasks cover different challenges in RL. Except for learning from **high-dimensional** observations, we include a **hard exploration** version of Cheetah Run by setting zero rewards below a reward threshold and applying an action penalty. We include Walker Run and Quadruped Walk for **complex dynamics**. We further include challenging manipulation tasks from MetaWorld.

Baselines In our experiments, we consider four different baselines for leveraging reward-free offline data. Note that as these baselines are not designed to handle multi-embodiment offline data, we preprocess the offline data to only contain task-related data for the baselines. Although this is an unfair comparison for WPT as baselines eliminate the difficulty raised with the large non-curated dataset, we still outperform these baselines by a large margin on tested tasks. The compared baselines are: *i) R3M*, which pre-trains visual representation with offline data, *ii) UDS*, which suggests labeling offline data with zero, *iii) ExPLORe*, which labels offline data with UCB reward, and *iv) JSRL-BC*, which collects online with both the training policy and a prior policy trained on offline data with behavior cloning.

Results Figure 2 shows comparison results with baselines. Our method outperforms or matches *all* compared baselines by a large margin. Compared to R3M, WPT stresses the importance of world model pre-training as well as reusing offline data during fine-tuning compared to representation learning only. R3M fails to improve sample efficiency on most of the tested tasks. The same conclusion is obtained in Hansen et al. (2023). UDS and ExPLORe reuse offline data by labeling offline data with zero rewards and UCB rewards respectively. The labeled offline data is concatenated with online data for off-policy updates. We find UDS only shows slightly better performance on the Walker Run task compared to R3M and JSRL-BC, showing the ineffectiveness of directly labeling offline data with zero rewards. ExPLORe shows better performance on 2/3 locomotion tasks compared to other baselines and shows signs of meaningful progress on two challenging manipulation tasks. However, WPT still outperforms ExPLORe by a large margin. Compared to ExPLORe, WPT demonstrates the superiority of world model pre-training as well as execution guidance.

Compared to JSRL-BC, our method also demonstrates a clear performance boost. JSRL-BC's performance heavily depends on the distribution of offline datasets. It can achieve good performance if a good prior actor can be extracted from the offline data. However, in our case, since we aim to lift the assumption of expert trajectories, the pre-trained BC agent usually fails to work well in the target task. As a result, JSRL-BC is only slightly better than the rest baselines on the Cheetah Run Hard, Assembly, and Shelf Place tasks while achieving similar results on the rest. In contrast, WPT works well on leveraging non-expert offline data. For example, the Quadruped Walk shows that WPT can nicely benefit from exploratory offline data collected by unsupervised RL, which enables it to control the Ant robot to walk forward from pixel inputs with only 100 trials.

In Fig. 2, we further compare our method with two widely used learning-from-scratch baselines, Dreamerv3 and DrQv2, on 72 tasks. We show that by leveraging the non-curated offline data, WPT can clearly improve the sample efficiency on a wide range of tasks. Furthermore, our method achieves promising performance on several hard exploration tasks, while learning-from-scratch baselines fail.

3 CONCLUSION

We propose WPT, a simple yet efficient approach to leverage non-curated offline data. We show that the generalist world model pre-trained on non-curated data can boost RL training spanning multiple tasks and multiple embodiments. Together with retrieval-based experience rehearsal and execution guidance, WPT outperforms baselines on a wide range of tasks, including 22 locomotion tasks and 50 manipulation tasks. WPT unlocks ample sources of offline data and our results show that leveraging this non-curated data leads to strong performance. Nevertheless, our method can be improved in multiple ways, for example, by extending our methods to real-world applications as well as proposing novel world model architectures.

Acknowledgments

We acknowledge CSC – IT Center for Science, Finland, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through CSC. We acknowledge the computational resources provided by the Aalto Science-IT project. We acknowledge funding from the Research Council of Finland (353138, 362407, 352788, 357301, 339730). Aidan Scannell was supported by the Research Council of Finland, Flagship program Finnish Center for Artificial Intelligence (FCAI).

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2004.

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos World Foundation Model Platform for Physical AI. In *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning Dexterous In-Hand Manipulation. *The International Journal of Robotics Research (IJRR)*, 2020.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient Online Reinforcement Learning with Offline Data. *International Conference on Machine Learning (ICML)*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning (CoRL)*, 2023a.

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. *Robotics: Science and Systems (RSS)*, 2023b.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, et al. Scaling Data-Driven Robotics with Reward Sketching and Batch Reinforcement Learning. In *Robotics: Science and Systems (RSS)*, 2020.

Jongseong Chae, Seungyul Han, Whiyoung Jung, Myungsik Cho, Sungho Choi, and Youngchul Sung. Robust Imitation Learning Against Variations in Environment Dynamics. In *International Conference on Machine Learning (ICML)*, 2022.

Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot Learning (CoRL)*, 2024.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss Library. *arXiv preprint arXiv:2401.08281*, 2024.

Benjamin Eysenbach, Vivek Myers, Sergey Levine, and Ruslan Salakhutdinov. Contrastive Representations Make Planning Easy. In *Advances in Neural Information Processing Systems Workshop (NeurIPS Workshop)*, 2023.

Scott Fujimoto and Shixiang Shane Gu. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. PWM: Policy Learning with Large World Models. *arXiv preprint arXiv:2407.02466*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces. *International Conference on Learning Representations (ICLR)*, 2022.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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 (ICML)*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. *International Conference on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. *International Conference on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*, 2023.

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. In *International Conference on Machine Learning (ICML)*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *International Conference on Learning Representations (ICLR)*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders are Scalable Vision Learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2309.17080*, 2023.

Ryan Julian, Benjamin Swanson, Gaurav S Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Never Stop Learning: The Effectiveness of Fine-Tuning in Robotic Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, 2020.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. MT-OPT: Continuous Multi-Task Robotic Reinforcement Learning at Scale. *arXiv preprint arXiv:2104.08212*, 2021.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Robotics: Science and Systems (RSS)*, 2024.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research*, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-Based Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment Anything. In *International Conference on Computer Vision (ICCV)*, 2023.

Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline Reinforcement Learning with Fisher Divergence Critic Regularization. In *International Conference on Machine Learning (ICML)*, 2021.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-Training for Robots: Offline RL Enables Learning New Tasks from a Handful of Trials. In *Robotics: Science and Systems (RSS)*, 2023.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble. In *Conference on Robot Learning (CoRL)*, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*, 2020.

Qiyang Li, Jason Zhang, Dibya Ghosh, Amy Zhang, and Sergey Levine. Accelerating Exploration with Unlabeled Prior Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and Opportunities in Offline Reinforcement Learning from Visual Observations. *Transactions on Machine Learning Research (TMLR)*, 2023.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are Sample-Efficient World Models. In *International Conference on Learning Representations (ICLR)*, 2023.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. *arXiv preprint arXiv:2006.09359*, 2020.

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.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Andrew Y Ng, Stuart Russell, et al. Algorithms for Inverse Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2000.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *International Conference on Robotics and Automation (ICRA)*, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 2019.

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.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-Driven Exploration by Self-Supervised Prediction. In *International Conference on Machine Learning (ICML)*, 2017.

Tim Pearce, Tabish Rashid, Dave Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. *arXiv preprint arXiv:2411.04434*, 2024.

Rafael Rafailov, Kyle Beltran Hatch, Victor Kolev, John D Martin, Mariano Phielipp, and Chelsea Finn. MOTO: Offline Pre-Training to Online Fine-tuning for Model-Based Robot Learning. In *Conference on Robot Learning (CoRL)*, 2023.

Sai Rajeswar, Pietro Mazzaglia, Tim Verbelen, Alexandre Piché, Bart Dhoedt, Aaron Courville, and Alexandre Lacoste. Mastering the Unsupervised Reinforcement Learning Benchmark from Pixels. In *International Conference on Machine Learning (ICML)*, 2023.

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. *Transactions on Machine Learning Research (TMLR)*, 2022.

Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville. Pretraining Representations for Data-Efficient Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Rutav Shah and Vikash Kumar. RRL: Resnet as Representation for Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2021.

Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling Diverse Vision Foundation Models for Robot Learning. In *Conference on Robot Learning (CoRL)*, 2024.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-End Robotic Reinforcement Learning without Reward Engineering. *Robotics: Science and Systems (RSS)*, 2019.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling Representation Learning from Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2021.

Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. SMART: Self-supervised Multi-task pretrAining with contRol Transformers. In *International Conference on Learning Representations*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An Open-Source Generalist Robot Policy. *Robotics: Science and Systems (RSS)*, 2024.

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.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al. Jump-Start Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster Level in StarCraft II using Multi-Agent Reinforcement Learning. *Nature*, 2019.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData V2: A Dataset for Robot Learning at Scale. In *Conference on Robot Learning (CoRL)*, 2023.

Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. VRL3: A Data-Driven Framework for Visual Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jialong Wu, Haoyu Ma, Chaoyi Deng, and Mingsheng Long. Pre-Training Contextualized World Models with In-the-Wild Videos for Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are Scalable World Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior Regularized Offline Reinforcement Learning. *arXiv preprint arXiv:1911.11361*, 2019.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked Visual Pre-Training for Motor Control. *arXiv preprint arXiv:2203.06173*, 2022.

Mengjiao Yang and Ofir Nachum. Representation Matters: Offline Pretraining for Sequential Decision Making. In *International Conference on Machine Learning (ICML)*, 2021.

Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't Change the Algorithm, Change the Data: Exploratory Data for Offline Reinforcement Learning. *arXiv preprint arXiv:2201.13425*, 2022.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-Based Offline Policy Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative Data Sharing for Multi-Task Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to Leverage Unlabeled Data in Offline Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2022.

Zishun Yu and Xinhua Zhang. Actor-Critic Alignment for Offline-to-Online Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2023.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-Trained Image Encoder for Generalizable Visual Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges. *IEEE Transactions on Cybernetics*, 2024.

Yanjie Ze, Nicklas Hansen, Yinbo Chen, Mohit Jain, and Xiaolong Wang. Visual Reinforcement Learning with Self-Supervised 3D Representations. *Robotics and Automation Letters*, 2023.

Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive Behavior Cloning Regularization for Stable Offline-to-Online Reinforcement Learning. *arXiv preprint arXiv:2210.13846*, 2022.

Yi Zhao, Le Chen, Jan Schneider, Quankai Gao, Juho Kannala, Bernhard Schölkopf, Joni Pajarinen, and Dieter Büchler. RP1M: A Large-Scale Motion Dataset for Piano Playing with Bimanual Dexterous Robot Hands. In *Conference on Robot Learning (CoRL)*, 2024.

APPENDICES

A METHOD

In this section, we detail our two-stage approach, which consists of *(i)* world model pre-training, which learns a generalist (i.e. multi-task & embodiment) world model, given offline data, which rather importantly, includes reward-free and non-expert data, and *(ii)* RL-based fine-tuning which leverages the pre-trained world model and online interaction in an offline-to-online fashion.

A.1 PROBLEM SETUP

In this paper, we assume the agent can access a static offline dataset $\mathcal{D}_{\text{off}}$, which consists of trajectories $\{\tau^i_{\text{off}}\}_{i=1}^{N_{\text{off}}}$, where τ^i_{off} includes observations and actions $\{o^i_t, a^i_t\}_{t=0}^T$ collected by *unknown* behavior policies. This means that *(i)* rewards r^i_t are unknown, *(ii)* $\mathcal{D}_{\text{off}}$ does not necessarily include expert trajectories, and *(iii)* datasets consist of multi-embodiment data. The agent interacts with the environment to collect labeled trajectories $\tau^i_{\text{on}} = \{o^i_t, a^i_t, r^i_t\}_{t=0}^T$ and stores them in an online dataset $\mathcal{D}_{\text{on}} = \{\tau^i_{\text{on}}\}_{i=1}^{N_{\text{on}}}$. The goal of this paper is to learn a high-performance policy from both $\mathcal{D}_{\text{off}}$ and $\mathcal{D}_{\text{on}}$ whilst minimizing the amount of online interaction (N_{on}) by leveraging the unlabeled offline data $\mathcal{D}_{\text{off}}$. Removing the requirement of rewards and expert trajectories enables the agent to leverage a large set of diverse datasets.

A.2 MULTI-EMBODIMENT WORLD MODEL PRE-TRAINING

During pre-training, instead of training one model per task – as done in previous work (Hafner et al., 2020; 2021; 2023; Hansen et al., 2024) – we show that a single multi-task & embodiment world model more effectively leverages the offline data. Compared to methods (Yang & Nachum, 2021; Schwarzer et al., 2021; Yuan et al., 2022; Parisi et al., 2022; Ze et al., 2023) that only pre-train representations, pre-training a world model creates a general understanding of the environment, which we show can further boost agents' performance during fine-tuning.

We adopt a widely used world model in the literature, the recurrent state space model (RSSM) (Hafner et al., 2019) by making several modifications: *(i)* we remove the task-related losses, *(ii)* we pad the action with zeros to unify the action dimension of different embodiments, and *(iii)* we scale the model size to 280M. Given these modifications, we show that RSSMs can successfully learn the dynamics of multiple embodiments and can be fine-tuned for mastering different tasks. Our first stage consists of pre-training the following components:

$$\begin{aligned} \text{Sequence model}: & \; h_t = f_\theta(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Encoder}: & \; z_t \sim q_\theta(z_t | h_t, o_t) \\ \text{Dynamics predictor}: & \; \hat{z}_t \sim p_\theta(z_t | h_t) \\ \text{Decoder}: & \; \hat{o}_t \sim d_\theta(\hat{o}_t | h_t, z_t). \end{aligned}$$

The models f_θ, q_θ, p_θ and d_θ are optimized jointly by minimizing:

$$\mathcal{L}(\theta) = \mathbb{E}_{p_\theta, q_\theta}\Big[\sum_{t=1}^{T} \underbrace{-\ln p_\theta(o_t|z_t, h_t)}_{\text{pixel reconstruction loss}} + \beta \cdot \underbrace{\mathrm{KL}\big(q_\theta(z_t|h_t, o_t) \;||\; p_\theta(z_t|h_t)\big)}_{\text{latent state consistency loss}}\Big]. \tag{1}$$

The first term minimizes the reconstruction error while the second term enables the latent dynamics learning. Note that we removed the task-related objectives, e.g., reward prediction and continue prediction. We note that there is plenty of room to improve the world model's pre-training, e.g., by leveraging recently developed self-supervised training methods (Eysenbach et al., 2023), or advanced architectures (Vaswani et al., 2017; Gu et al., 2022). These improvements are orthogonal to our method and we leave it as future work.

A.3 RL-BASED FINE-TUNING

In our fine-tuning stage, we allow the agent to interact with the environment to collect new data $\tau^i_{\text{on}} = \{o^i_t, a^i_t, r^i_t\}_{t=0}^T$. The newly collected data is used to learn a reward function via supervised learning whilst also fine-tuning the world model with Eq. (1). For simplicity, we represent the concatenate of h_t and z_t as $s_t = [h_t, z_t]$. The actor and critic are trained with imagined trajectories $\tilde{\tau}(s, a)$ generated by rolling out the policy $\pi_\phi(a|s)$ in the world model p_θ starting from the initial state distribution $p_0(s)$, which are samples from replay buffer. Following DreamerV3, the critic $v_\phi(V_t^\lambda \mid s_t)$ learns to approximate the distribution over the λ-return V_t^λ calculated as,

$$\underbrace{V_t^\lambda}_{\lambda-\text{return}} = \hat{r}_t + \gamma \begin{cases} (1-\lambda)v_{t+1}^\lambda + \lambda V_{t+1}^\lambda & \text{if } t < H \\ v_H^\lambda & \text{if } t = H \end{cases}, \tag{2}$$

where $v_t^\lambda = \mathbb{E}[v_\phi(V_t^\lambda \mid s_t)]$ denotes the expected value of the distribution predicted by the critic. It is trained using maximum likelihood by minimizing

$$\mathcal{L}(v_\phi) = \mathbb{E}_{p_\theta, \pi_\phi}\left[-\sum_{t=1}^{H-1} \ln v_\phi(V_t^\lambda \mid s_t)\right]. \tag{3}$$

The actor $\pi_\phi(a_t|s_t)$ is then updated by maximizing the λ-return regularized by actor entropy $\mathbf{H}[a_t|s_t]$:

$$\mathcal{L}(\pi_\phi) = \mathbb{E}_{p_\theta, \pi_\phi}\left[\sum_{t=1}^{H-1} \left(-v_t^\lambda - \eta \cdot \mathbf{H}[a_t|s_t]\right)\right]. \tag{4}$$

In practice, we found that using the pre-trained world model alone fails to work well in many cases, especially for hard-exploration tasks. To identify the reasons, we need to rethink the role of each component in policy updates. Given the policy update rule in Eq. (4), the policy update is infected by the distribution of imagined trajectories $\tilde{\tau}(s, a) = p_0(s) \prod_{t=0}^{H-1} \pi_\phi(a_t|s_t) p_\theta(s_{t+1}|s_t, a_t)$ and the reward model $r_\theta(s_t, a_t)$, i.e., the initial state distribution $p_0(s)$, the policy $\pi_\phi(a|s)$, the dynamics model $p_\theta(s_{t+1}|s_t, a_t)$, and the reward model $r_\theta(s_t, a_t)$. Intuitively, as shown in Fig. 3, the policy will learn a good policy if the imagined trajectories $\tilde{\tau}$ contain "promising" data for policy learning and the reward function $r_\theta(s_t, a_t)$ can correctly label those trajectories. The world model pre-training gives a better initialization of $p_\theta(s_{t+1}|s_t, a_t)$. However, when the online data distribution is narrow – as in hard exploration tasks – the initial state distribution $p_0(s)$ can also be narrow, which prevents the actor from reaching high-reward states during imagination. Also, the reward function $r_\theta(s, a)$ trained on the narrow online data distribution will incorrectly assign rewards for transitions generated by the world model pre-trained on a broader distribution. That is, the agent fails to improve the policy even if "promising" trajectories exist in $\tilde{\tau}$, since incorrect rewards are assigned. Furthermore, fine-tuning the world model $p_\theta(s_{t+1}|s_t, a_t)$ on the narrow distribution causes catastrophic forgetting. In the following, we improve the agents' performance by reusing non-curated offline data for *i)* augmenting the initial state distribution $p_0(s)$, *ii)* collecting diverse online trajectories, and *iii)* improving the reward predictions. See Fig. 3 for an illustrative motivation.

Retrieval-based Experience Rehearsal Experience replay is a simple yet effective approach to prevent catastrophic forgetting (Khetarpal et al., 2022) as well as improving policy learning (Ball et al., 2023; Li et al., 2023). In the RL setting, Ball et al. (2023) demonstrates promising results by replaying offline data to improve sample efficiency but requires the offline data to be reward-labeled. ExPLORe (Li et al., 2023) reuses single-embodiment but reward-free offline data to encourage exploration by labeling offline data with UCB rewards. Unlike the replay scheme used in ExPLORe, where the offline datasets are relatively small and well-structured, directly replaying the non-curated offline data can be challenging. Our dataset can be $\sim 100\times$ larger and consist of different tasks and embodiments. As such, directly labeling the offline dataset with UCB reward or even replaying the dataset can be infeasible.

To enable the usage of the non-curated offline data, we propose a simple retrieval-based approach to filter the trajectories that are close to the downstream tasks. Specifically, we retrieve a subset of trajectories from the non-curated offline data based on the neural feature distance between the online samples and the trajectories in the offline dataset. We calculate the distance as :

$$\mathbf{D} = ||\mathrm{e}_\theta(o_{\text{on}}) - \mathrm{e}_\theta(o_{\text{off}})||_2, \tag{5}$$

where e_θ is a feature extractor and o_{on} and o_{off} are images from the initial observation of trajectories from the online buffer and the offline dataset, respectively. In practice, we found that neural representation matters for retrieval, and the encoder learned via world model pre-training works better than the general visual model R3M (Nair et al., 2022).

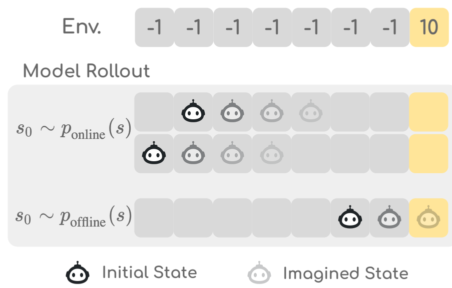

Figure 3: Motivation of using experience rehearsal and execution guidance. During fine-tuning, experience rehearsal augments the initial state $p_0(s)$ used for model rollout, which enables the model rollout to start from informative states. Execution guidance helps the agent to collect data that is close to the offline data distribution, which enriches the diversity of the collected data and improves the reward model's accuracy on a broader state distribution.

To efficiently search over the whole dataset, we pre-build the key-value pairs between the trajectories ID and its neural features and use Faiss (Douze et al., 2024) for similarity search. Once the key-value database is built, the buffer retrieval procedure only takes seconds. The retrieved offline buffer, although reward-free, contains embodiment-specific data and is feasible to use. The retrieved offline data can be used to prevent catastrophic forgetting in the world model and to augment the initial state distribution $p_0(s)$ used for trajectory imagination.

Execution Guidance via Prior Actors The common practice in RL training is to initialize the replay buffer with a random policy and gradually collect new data by interacting with the environment using the training policy. When using a pre-trained world model, it is preferred to start the training from a distribution that is close to the offline data distribution. This has three benefits: First, offline data usually contains useful information for policy training, such as near-expert trajectories or a broader state-action distribution. Second, the distribution shift between the offline and online data may ruin the pre-trained weights, so bringing the initial distribution close to the offline data is a good strategy to prevent this. Thirdly, the reward function can get better predictions for states that are close to the offline data distribution. This improves policy learning as the initial states used to generate the imagined trajectories $\tilde{\tau}$ are augmented with offline samples.

To guide the online data collection such that it remains close to the offline data distribution, we simply train a prior actor π_{bc} via behavior cloning on the *retrieved* buffer. During online data collection, we switch between the policy prior π_{bc} and the RL agent π_ϕ according to a pre-defined schedule. Specifically, at the beginning of an episode, we decide whether to use π_{bc} according to the scheduled probability. If the π_{bc} is selected to use, we then randomly select the starting time step t_{bc} and duration H of using π_{bc}. π_ϕ is then used for the rest of the time steps. This procedure is similar to JSRL (Uchendu et al., 2023) but with a couple of differences: *i)* WPT leverages non-curated offline data while JSRL uses task-specific offline data and *ii)* WPT shows the superiority of a model-based approach for leveraging offline data while JSRL is a model-free approach. As a result, we show that WPT obtains better performance than JSRL on tested tasks. The full algorithm is presented in Alg. 1.

Algorithm 1 Generalist World Model Pre-Training for Efficient RL

Require: Non-curated offline data $\mathcal{D}_{\text{off}}$, Online data $\mathcal{D}_{\text{on}} \leftarrow \emptyset$, Retrieval data $\mathcal{D}_{\text{retrieval}} \leftarrow \emptyset$
World model $f_\theta, q_\theta, p_\theta, d_\theta$
Policy $\pi_{\phi_{\text{RL}}}, \pi_{\phi_{\text{BC}}}$, Value function v_ϕ and Reward r_ξ.

// Task-Agnostic World Model Pre-Training
for num. pre-train steps **do**
  Randomly sample mini-batch $\mathcal{B}_{\text{off}} : \{o_t, a_t, o_{t+1}\}_{t=0}^{T}$ from $\mathcal{D}_{\text{off}}$.
  Update world model $f_\theta, q_\theta, p_\theta, d_\theta$ by minimizing Eq. (1) on sampled batch $\mathcal{B}$.
end for

// Task-Specific Training
// Experience Retrieval
Collect one initial observation o_{on}^{0} from the environment.
Compute the visual similarity between o_{on} and initial observations of trajectories o_{off} in $\mathcal{D}_{\text{off}}$ using Eq. (5).
Select R trajectories according to Eq. (5) and fill $\mathcal{D}_{\text{retrieval}}$.

// Behavior Cloning Policy Training
for num. bc updates **do**
  Randomly sample mini-batch $\mathcal{B}_{\text{retrieval}} : \{o_i, a_i\}_{i=0}^{N}$ from $\mathcal{D}_{\text{retrieval}}$.
  Update $\pi_{\phi_{\text{BC}}}$ by minimizing $-\frac{1}{N}\sum_{i=0}^{N} \log \pi_{\phi_{\text{BC}}}(a_t|o_t)$.
end for

// Task-Specific RL Fine-Tuning
for num. episodes **do**
  // Collect Data
  Decide whether to use $\pi_{\phi_{\text{BC}}}$ according to the predefined schedule.
  if Select $\pi_{\phi_{\text{BC}}}$ **then**
    Randomly select the starting time step k and the rollout horizon H.
  end if
  $t \leftarrow 0$
  while $t \leq$ episode length **do**
    $a_t = \pi_{\phi_{BC}}(a_t|o_t)$ if Use $\pi_{\phi_{\text{BC}}}$ and $k \leq t \leq H$ else $a_t = \pi_{\phi_{RL}}(a_t|o_t)$.
    Interact the environment with a_t. Store $\{o_t, a_t, r_t, o_{t+1}\}$ to $\mathcal{D}_{\text{on}}$.
    $t \leftarrow t + 1$
  end while

  // Update Models
  for num. grad steps **do**
    Randomly sample mini-batch $\mathcal{B}_{\text{on}} : \{o_t, a_t, r_t, o_{t+1}\}_{t=0}^{T}$ from $\mathcal{D}_{on}$ and $\mathcal{B}_{\text{retrieval}} : \{o_t, a_t, r_t, o_{t+1}\}_{t=0}^{T}$ from $\mathcal{D}_{\text{retrieval}}$.
    Update world model $f_\theta, q_\theta, p_\theta, d_\theta$ by minimizing Eq. (1) on sampled batch $\{\mathcal{B}_{\text{on}}, \mathcal{B}_{\text{retrieval}}\}$.
    Update r_ξ by minimizing $-\frac{1}{N}\sum_{i=0}^{N} \log p_\xi(r_t|s_t)$ on $\mathcal{B}_{\text{on}}$. $\triangleleft\ s_t = [h_t, z_t]$
    // Update policy and value function
    Generate imaginary trajectories $\tilde{\tau} = \{s_t, a_t, s_{t+1}\}_{t=0}^{T}$ by rolling out h_θ, p_θ with policy $\pi_{\phi_{\text{RL}}}$.
    Update policy $\pi_{\phi_{\text{RL}}}$ and value function v_ϕ with Eq. (4).
  end for
end for

B MORE RESULTS

In this section, we present WPT's performance on task adaptation, which is particularly important to empower the agent's life-long learning ability. Furthermore, we conduct an ablation study to investigate the role of proposed design choices.

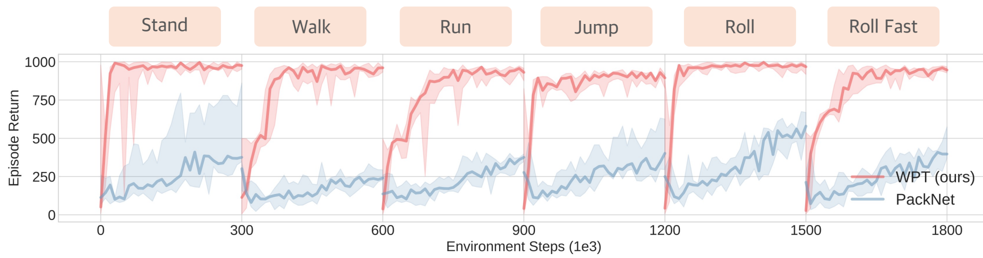

Figure 4: WPT enables fast task adaptation. We train an RL agent to control an Ant robot from DMControl to finish a series of tasks incrementally. WPT outperforms the widely used baseline by a decent margin by properly leveraging non-curated offline data.

B.1 WPT Enables Fast Task Adaptation

In many real-world applications, we want the robot to continuously adapt to new tasks. In this section, we investigate whether WPT benefits task adaptation. In this setting, the agent is required to incrementally solve a sequence of tasks. The setting is very similar to continual reinforcement learning (CRL) or life-long RL (Parisi et al., 2019; Khetarpal et al., 2022) but with a limited set of tasks for simplicity. Note that CRL has a broad scope; assumptions and experiment setups vary among methods, which makes it difficult to set up a fair comparison with other methods. Instead of proposing a SOTA CRL method, we aim to demonstrate that our method offers a simple yet general recipe to leverage previous data that also fits the CRL setting.

Setup & baselines We set our continual multi-task adaption experiment based on the Ant robot from the DeepMind Control Suite. Specifically, the agent needs to sequentially solve stand, walk, run, jump, roll, and roll fast tasks. We train 300K environment steps for each task. To solve one task, the agent accesses all previous experiences as well as the model's weights trained on previous tasks. To have a fair comparison, i.e., having comparable model parameters and eliminating the potential effects from pre-training on other tasks, we pre-train a small world model only on the Ant domain. We compare our method with PackNet (Mallya & Lazebnik, 2018), which is a common baseline for continual learning. PackNet iteratively prunes the actor's parameters by keeping the weights with a larger magnitude while re-initializing the rest. By doing so, the actor maintains skills learned previously, which may help the learning of new tasks. For each new task, we fine-tune the actor model by iterative pruning while randomly initializing the critic model as rewards are not shared among tasks.

Results Figure 4 shows policy performance on incremental task adaption. As we can see WPT outperforms baselines by a large margin. Compared to PackNet, WPT shows great potential for pre-training of world models with non-curated offline data together with experience rehearsal and actor prior. For the tested Ant domain, WPT enables the agent to adapt to new tasks within only 100 trials on all six tasks. With the limited sample budget, PackNet only achieves $\sim 20 - 60\%$ episodic returns of WPT. We would argue the diversity of the non-curated offline data contributes to WPT's supreme performance on task adaptation.

B.2 Ablations

In this section, we investigate the role of each component of our method. We use the same set of tasks used in Fig. 2. As shown in Fig. 5, world model pre-training shows promising results when the offline dataset consists of diverse trajectories, such as data collected by exploratory agents, while fails to work well when the offline data distribution is relatively narrow as in the Meta-World tasks. This is due to *i)* world model pre-training alone failing to fully leverage the rich action prior in the offline data and *ii)* the distributional shift between offline and online data hurts the world model fine-tuning. The proposed retrieval-based experience rehearsal and execution guidance helps the agent benefit from action prior in the offline data and enriches the online sample diversity, which together enables WPT to achieve strong performance on a wide range of tasks.

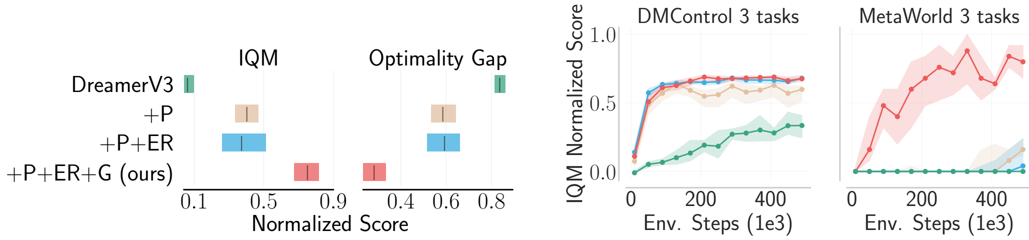

Figure 5: Ablation study on the role of each component. "P" represents world model pretraining, "ER" means experience rehearsal, and "G" represents execution guidance. Together with the proposed retrieval-based experience rehearsal and execution guidance, world model pre-training boosts RL performance on a wide range of tasks.

C Related Work

In this section, we review methods which leverage offline data, including pre-training in the context of RL and the so-called generalist agents. See Table 1 for a comparison of what types of data the methods can use.

RL with task-specific offline datasets Leveraging offline data is a promising direction to improve the sample efficiency in RL. One representative method is Offline RL. Offline RL trains agents from offline data without interacting with the environment. It usually constrains the distance between the learned policy and behavior policies in different ways (Kumar et al., 2019; 2020; Wu et al., 2019; Kostrikov et al., 2021; 2022; Fujimoto & Gu, 2021; Uchendu et al., 2023). However, the performance of the learned policy is highly affected by the quality of offline datasets (Yarats et al., 2022). To continue improving the policy, offline-to-online RL (Nair et al., 2020; Zhao et al., 2022; Lee et al., 2022; Yu & Zhang, 2023; Rafailov et al., 2023) was proposed.

Offline RL and offline-to-online RL commonly suffer from training instability and require additional treatments (Lee et al., 2022; Lu et al., 2023). RLPD (Ball et al., 2023) shows that off-policy RL delivers strong performance by directly concatenating the offline data with online data. However, RLPD still requires reward-labeled task-specific offline data and hasn't discussed the multi-embodiment cases. Recently, ExPLORe (Li et al., 2023) labels reward-free offline data with approximated upper confidence bound (UCB) of rewards to solve hard exploration tasks. However, ExPLORe obtains good performance when the offline data includes near-expert data of the target tasks, while we consider a more general setting with non-curated offline data.

RL with multi-task offline datasets Recently, several works have applied offline RL in a multi-task setting (Kalashnikov et al., 2021; Yu et al., 2021; Kumar et al., 2023; Julian et al., 2020; Hansen et al., 2024), but require known rewards. Georgiev et al. (2024) pre-trains a world model for multi-task RL but is designed for state-based inputs and requires reward-labeled offline data for world model pre-training.

To lift the assumption of known rewards, human labeling (Cabi et al., 2020; Singh et al., 2019), inverse RL (Ng et al., 2000; Abbeel & Ng, 2004) or generative adversarial imitation learning (Ho & Ermon, 2016) can be applied. However, this requires human labor or expert demonstrations. Yu et al. (2022) directly sets zero rewards for unlabelled data, which introduces additional bias.

Another stream of work leverages in-the-wild data for RL training. Most methods focus on representation learning. Stooke et al. (2021); Yang & Nachum (2021); Schwarzer et al. (2021); Shah & Kumar (2021); Yuan et al. (2022); Wang et al. (2022); Parisi et al. (2022); Sun et al. (2023); Ze et al. (2023) investigate different representation learning methods and show promising performance boosts during fine-tuning.

Generalist Agents RL methods usually perform well on a single task (Vinyals et al., 2019; Andrychowicz et al., 2020), however, this contrasts with humans that can perform multiple tasks well. Recent works have proposed generalist agents that master a diverse set of tasks with a single agent (Reed et al., 2022; Brohan et al., 2023b; Team et al., 2024; Zhao et al., 2024). These methods typically resort to scalable models and large datasets and are trained via imitation learning (Brohan et al., 2023b;a; O'Neill et al., 2023; Khazatsky et al., 2024). In contrast, we seek to train a generalist world model and use it to boost RL performance for multiple tasks and embodiments.

World models World models learn to predict future observations or states based on historical information. World models have been widely investigated in online model-based RL (Ha & Schmidhuber, 2018; Hafner et al., 2020; Micheli et al., 2023; Alonso et al., 2024). Recently, the community has started investigating scaling world models (Ha & Schmidhuber, 2018), for example, Hu et al. (2023); Pearce et al. (2024); Wu et al. (2025); Agarwal et al. (2025) train world model with Diffusion Models or Transformers. However, these models are usually trained on demonstration data. In contrast, we explore the offline-to-online RL setting – closely fitting the successful pre-train and then fine-tune paradigm – and we focus on leveraging reward-free and non-expert data to increase the amount of available data for pre-training.

D HYPERPARAMETERS

In this section, we list important hyperparameters used in WPT.

Table 2: Hyperparameters used in WPT.

Hyperparameter	**Value**
Pre-training	
Stacked images	1
Pretrain steps	200,000
Batch size	16
Sequence length	64
Replay buffer capacity	Unlimited
Replay sampling strategy	Uniform
RSSM	
Hidden dimension	12288
Deterministic dimension	1536
Stochastic dimension	32 * 96
Block number	8
Layer Norm	True
CNN channels	[96, 192, 384, 768]
Activation function	SiLU
Optimizer	
Optimizer	Adam
Learning rate	1e-4
Weight decay	1e-6
Eps	1e-5
Gradient clip	100
Fine-tuning	
Warm-up frames	15000
Execution Guidance Schedule	linear(1,0,50000) for DMControl linear(1,0,1,150000) for Meta-Wolrd
Action repeat	2
Offline data mix ratio	0.25
Discount	0.99
Discount lambda	0.95
MLPs	[512, 512, 512]
MLPs activation	SiLU
Actor critic learning rate	8e-5
Actor entropy coef	1e-4
Target critic update fraction	0.02
Imagine horizon	16

E Full Results

In Table 3 and Table 4, we list the success rate of 50 Meta-World benchmark tasks with pixel inputs. In Table 5, we list the episodic return of DMControl of 22 tasks. We compare WPT at 150k samples with two widely used baselines Dreamer v3 and DrQ v2 at both 150k samples and 1M samples. We mark the best result with a bold front at 150k samples and use underlining to mark the highest score overall.

E.1 Meta-World Benchmark

Table 3: Success rate of Meta-World benchmark with pixel inputs.

Tasks	Dreamer v3 @ 1M	DrQ v2 @ 1M	Dreamer v3 @ 150k	DrQ v2 @ 150k	WPT (ours) @ 150k
Assembly	0.0	0.0	0.0	0.0	<u>**0.2**</u>
Basketball	0.0	<u>0.97</u>	0.0	0.0	**0.4**
Bin Picking	0.0	<u>0.93</u>	0.0	0.33	**0.8**
Box Close	0.13	<u>0.9</u>	0.0	0.0	<u>**0.9**</u>
Button Press	<u>1.0</u>	0.7	0.47	0.13	**0.9**
Button Press Topdown	<u>1.0</u>	<u>1.0</u>	0.33	0.17	<u>**1.0**</u>
Button Press Topdown Wall	<u>1.0</u>	<u>1.0</u>	0.73	0.63	<u>**1.0**</u>
Button Press Wall	<u>1.0</u>	<u>1.0</u>	0.93	0.77	<u>**1.0**</u>
Coffee Button	1.0	1.0	1.0	1.0	1.0
Coffee Pull	0.6	<u>0.8</u>	0.0	**0.6**	**0.6**
Coffee Push	0.67	<u>0.77</u>	0.13	0.2	**0.7**
Dial Turn	<u>0.67</u>	0.43	0.13	0.17	<u>**0.67**</u>
Disassemble	0.0	0.0	0.0	0.0	0.0
Door Close	-	-	-	-	1.0
Door Lock	<u>1.0</u>	0.93	0.6	**0.97**	0.9
Door Open	<u>1.0</u>	0.97	0.0	0.0	**0.8**
Door Unlock	1.0	1.0	<u>**1.0**</u>	0.63	0.8
Drawer Close	0.93	1.0	0.93	<u>**1.0**</u>	0.9
Drawer Open	0.67	0.33	0.13	0.33	<u>**1.0**</u>
Faucet Open	1.0	1.0	0.47	0.33	<u>**1.0**</u>
Faucet Close	0.87	1.0	<u>**1.0**</u>	<u>**1.0**</u>	0.8
Hammer	1.0	1.0	0.07	0.4	<u>**1.0**</u>
Hand Insert	0.07	<u>0.57</u>	0.0	0.1	**0.4**
Handle Press Side	1.0	1.0	1.0	1.0	<u>**1.0**</u>
Handle Press	1.0	1.0	0.93	0.97	<u>**1.0**</u>
Handle Pull Side	0.67	1.0	0.67	0.6	<u>**1.0**</u>
Handle Pull	0.67	0.6	0.33	0.6	<u>**1.0**</u>
Lever Pull	0.73	<u>0.83</u>	0.0	0.33	**0.8**

Table 4: Success rate of Meta-World benchmark with pixel inputs (Cont.).

Tasks	Dreamer v3 @ 1M	DrQ v2 @ 1M	Dreamer v3 @ 150k	DrQ v2 @ 150k	WPT (ours) @ 150k
Peg Insert Side	1.0	1.0	0.0	0.27	<u>**1.0**</u>
Peg Unplug Side	<u>0.93</u>	0.9	0.53	0.5	**0.8**
Pick Out of Hole	0.0	0.27	0.0	0.0	<u>**0.3**</u>
Pick Place Wall	0.2	0.17	0.0	0.0	<u>**0.5**</u>
Pick Place	<u>0.67</u>	<u>0.67</u>	0.0	0.0	**0.2**
Plate Slide Back Side	1.0	1.0	0.93	1.0	<u>**1.0**</u>
Plate Slide Back	1.0	1.0	0.8	0.97	<u>**1.0**</u>
Plate Slide Side	<u>1.0</u>	0.9	**0.73**	0.5	0.5
Plate Slide	1.0	1.0	0.93	1.0	<u>**1.0**</u>
Push Back	<u>0.33</u>	<u>0.33</u>	0.0	0.0	**0.2**
Push Wall	0.33	0.57	0.0	0.0	<u>**0.9**</u>
Push	0.26	<u>0.93</u>	0.0	0.13	**0.7**
Reach	<u>0.87</u>	0.73	**0.67**	0.43	0.3
Reach Wall	<u>1.0</u>	0.87	0.53	0.7	**0.9**
Shelf Place	0.4	0.43	0.0	0.0	<u>**0.87**</u>
Soccer	0.6	0.3	0.13	0.13	<u>**0.67**</u>
Stick Push	0.0	0.07	0.0	0.0	<u>**0.4**</u>
Stick Pull	0.0	0.33	0.0	0.0	<u>**0.67**</u>
Sweep Into	0.87	<u>1.0</u>	0.0	0.87	**0.9**
Sweep	0.0	<u>0.73</u>	0.0	0.3	**0.6**
Window Close	1.0	1.0	0.93	<u>**1.0**</u>	0.8
Window Open	1.0	0.97	0.6	<u>**1.0**</u>	0.9
Mean	<u>0.900</u>	0.753	0.360	0.442	**0.750**
Medium	0.870	<u>0.900</u>	0.130	0.330	**0.800**

E.2 DMControl Benchmark

Table 5: Episodic return of DMControl benchmark with pixel inputs.

Tasks	Dreamer v3 @ 500k	DrQ v2 @ 500k	Dreamer v3 @ 150k	DrQ v2 @ 150k	WPT (ours) @ 150k
CartPole Balance	994.3	992.3	955.8	983.3	**995.0**
Acrobot Swingup	222.1	30.3	**85.2**	20.8	50.3
Acrobot Swingup Sparse	2.5	1.17	1.7	1.5	**26.9**
Acrobot Swingup Hard	-0.2	0.3	2.0	0.4	**10.7**
Walker Stand	965.7	947.6	946.2	742.9	**969.4**
Walker Walk	949.2	797.8	808.9	280.1	**959.1**
Walker Run	616.6	299.3	224.4	143.0	**728.0**
Walker Backflip	293.6	96.7	128.2	91.7	**306.0**
Walker Walk Backward	942.9	744.3	625.9	470.9	**863.6**
Walker Walk Hard	-2.1	-9.5	-4.7	-17.1	**878.3**
Walker Run Backward	363.8	246.0	229.4	167.4	**349.3**
Cheetah Run	843.7	338.1	**621.4**	251.2	526.1
Cheetah Run Front	473.8	202.4	143.1	108.4	**360.5**
Cheetah Run Back	657.4	294.4	407.6	171.2	**446.0**
Cheetah Run Backwards	693.8	384.3	**626.6**	335.6	542.2
Cheetah Jump	597.0	535.6	200.8	251.8	**634.1**
Quadruped Walk	369.3	258.1	145.2	76.5	**933.6**
Quadruped Stand	746.0	442.2	227.2	318.9	**936.4**
Quadruped Run	328.1	296.5	183.0	102.8	**802.5**
Quadruped Jump	689.6	478.3	168.3	190.5	**813.5**
Quadruped Roll	663.9	446.0	207.9	126.2	**970.8**
Quadruped Roll Fast	508.8	366.9	124.8	164.7	**782.0**
Mean	541.81	372.23	320.86	226.49	**631.10**
Medium	606.8	318.70	204.35	166.05	**755.0**

F Task Visualization

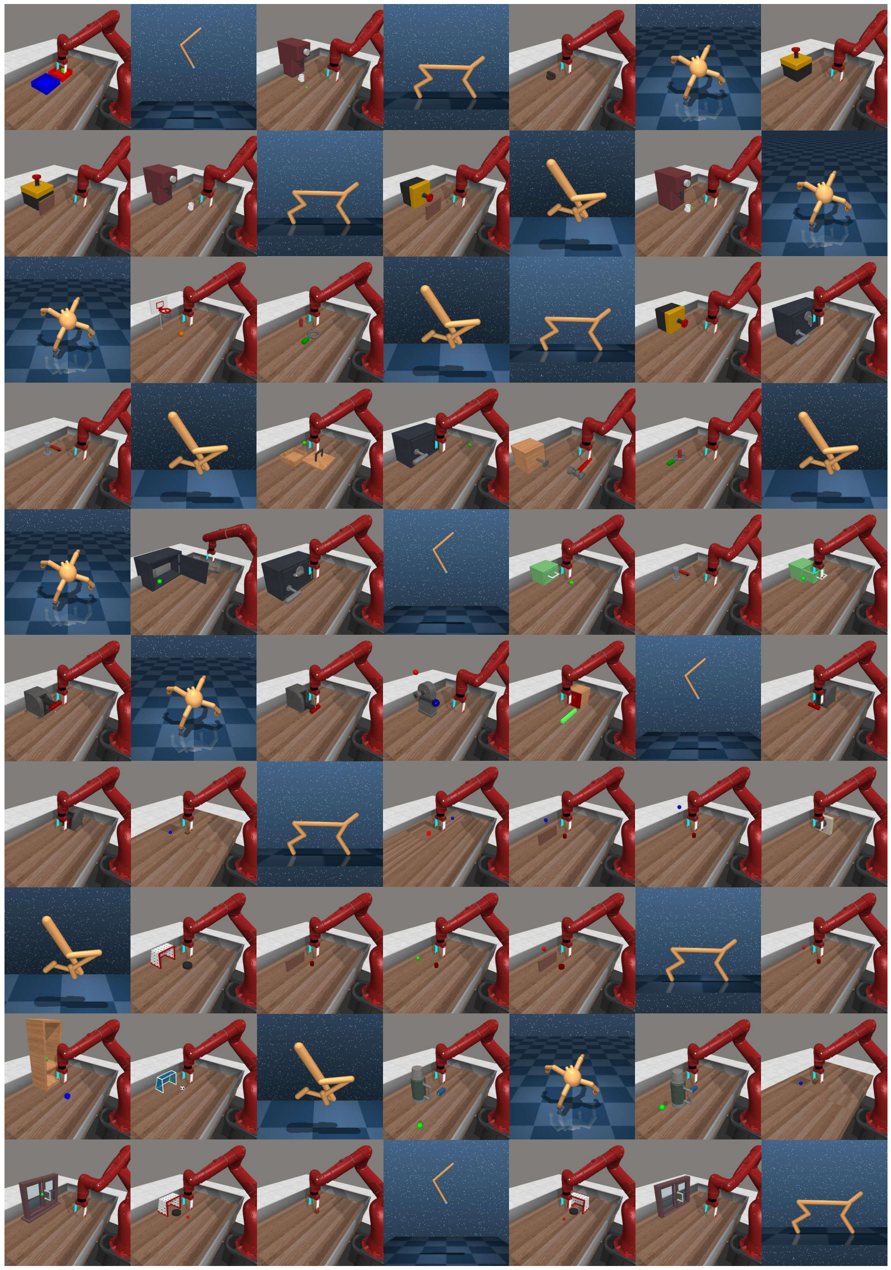

Figure 6: Visualization of tasks from DMControl and Meta-World used in our paper.