Agent-Centric Human Demonstrations Train World Models

James Staley
james.staley625703@tufts.edu
College of Engineering
Tufts University

Shivam Goel
shivam.goel@tufts.edu
College of Engineering
Tufts University

Yash Shukla
yash.shukla@tufts.edu
College of Engineering
Tufts University

Elaine Schaertl Short
elaine.short@tufts.edu
Tufts University
Assistant Professor, Computer Science
Clare Boothe Luce Professorship in Engineering

Abstract

Previous work in interactive reinforcement learning considers human behavior directly in agent policy learning, but this requires estimating the *distribution* of human behavior over many samples to prevent bias. Our work shows that model-based systems can avoid this problem by using small amounts of human data to guide world-model learning rather than agent-policy learning. We show that this approach learns faster and produces useful policies more reliably than prior state-of-the-art. We evaluate our approach with expert human demonstrations in two environments: PinPad5, a fully observable environment that prioritizes task composition, and MemoryMaze, a partially observable environment that prioritizes exploration and memory. We show an order of magnitude speed-up in learning and reliability with only nine minutes of expert human demonstration data.

1 Introduction

Goals for agents in reinforcement learning (RL) can often easily be described by specific world state conditions, but it can take a prohibitively long time for an agent to discover a task's goal state, even with exploration rewards. In this context, using guidance from a human teacher can substantially speed learning. One commonly used method of incorporating human guidance is *imitation learning*. Standard approaches to imitation learning shape an agent's behavior either directly through its policy or by modeling a human's dense reward function. These approaches may include directly incorporating demonstrations (Ross et al., 2011; Kelly et al., 2019; Spencer et al., 2020); learning to distinguish expert-actions from policy-actions (Ho & Ermon, 2016; Rafailov et al., 2021); training policies using human demonstrations as labels (Bain & Sammut, 1995; Torabi et al., 2018); learning reward functions from scalar feedback (Knox & Stone, 2009; Warnell et al., 2018) or preferences (Wirth et al., 2016; Bai et al., 2022); or learning to explore(Villasevil et al., 2023). These methods speed up learning, but typically create models from human demonstrations with limited state-action coverage, causing them to fall short when applied to real-world distributions. Estimating a real-world state-action or reward distribution in an unbiased way requires more data than a single person can reasonably be expected to provide. The goal of our work is to address this limitation by enabling a single human to guide an agent to learn tasks on human-relevant timescales (e.g. from periodic demonstrations within one work-week).

We build on recent works in model-based reinforcement learning (MBRL), which use a world model to train an agent's policy (Moerland et al., 2023). MBRL increases sample efficiency to enable long-horizon task learning with few human demonstrations. In this approach, the world model itself can be taught about a task's sparse reward without directly shaping an agent's policy or reward function. World models can learn from trajectories produced by any policy, so they can be guided early in training by human demonstration data without significant modification. *Intuitively, by never explicitly considering the human state-visitation density or action-selection likelihood we prevent poor modeling of human behavior.* Furthermore, this approach avoids problems of distribution shift in two ways. First, directly controlling the agent using its own observations and affordances avoids shifts between the human and agent observation and action spaces. Second, avoiding updating the agent's policy or reward function directly from human demonstrations prevents problems caused by insufficient coverage of the state-action space, which otherwise could result in a brittle agent policy (Ross et al., 2011; Rajeswaran et al., 2017). Instead, the agent learns on-policy from the world model without ever having to consider the human's behavior distribution.

In this paper, we propose teaching the world model in MBRL as an effective form of Learning from Demonstration (LfD). We demonstrate the effectiveness of this approach in two simulated environments with sparse reward: PinPad5, a long-horizon fully-observable image observation task requiring a precisely composed series of states, and MemoryMaze, a partially observable image observation task focused on memory and exploration. We build on DreamerV3 (Hafner et al., 2023), which trains an actor-critic on-policy purely from the model's imagined unrolls. Human demonstrations inform the world model's learning, which in turn guides the agent's learning. Dreamer is particularly well-suited to incorporate human demonstrations because of its state-of-the-art performance on RL tasks, allowing us to more easily isolate the impact of human demonstrations. We show that this approach substantially improves the speed and consistency of learning. With nine to eighteen minutes of human intervention, we attain 90% of max reward four to six times faster and more consistently than Dreamer or any baseline.

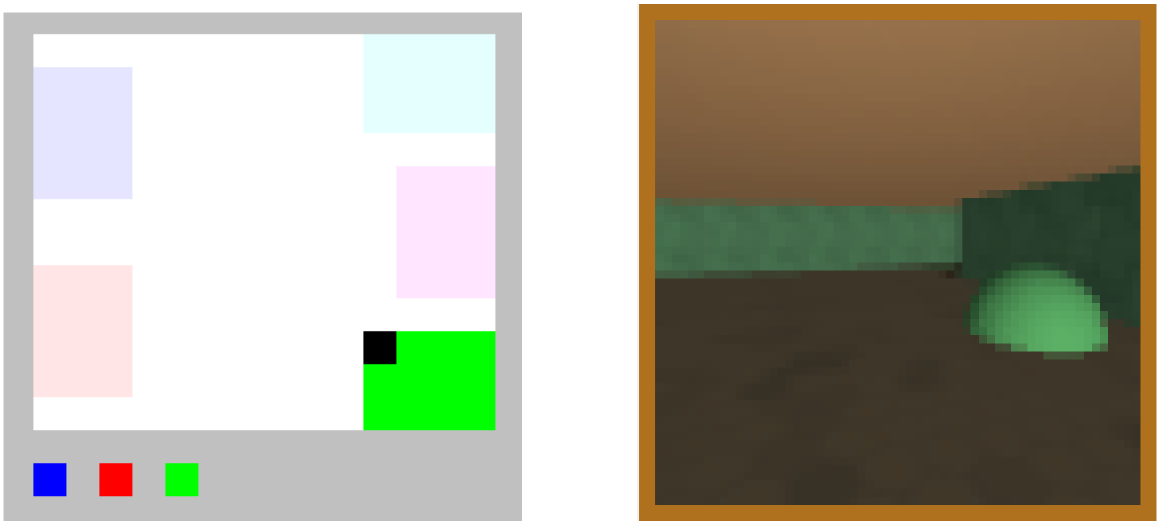
Overall, a focus on training world models rather than agent policies or reward functions from human demonstrations opens promising new directions for research in human-in-the-loop learning. In this paper, we show that this is effective in environments with sparse, positive rewards, but this approach may also be helpful in understanding harmful or preferred states. In this way, the learning agent is able to extract the most important information from human demonstrations while remaining robust to noise and errors in those demonstrations.

2 Background

2.1 Learning from Demonstration and Interactive Imitation Learning

Interactive Imitation Learning (IIL) uses a human or pre-trained oracle to guide agent behavior within the learning environment by offering corrections through provided feedback (Knox & Stone, 2009; Warnell et al., 2018), comparisons (Wirth et al., 2016; Bai et al., 2022), or demonstration (Ross et al., 2011; Kelly et al., 2019; Spencer et al., 2020). IIL operates over a distribution induced by the learner, rather than expert, which can improve sample efficiency by offering a natural and intuitive teaching approach for non-experts, and reducing distributional mismatch / covariate shift(Celemin et al., 2022).

Learning from Demonstration (LfD) is a form of Interactive Imitation Learning (IIL) that incorporates information from human demonstrations to speedup learning or customize agent behavior. The resulting demonstrations can be treated as a dataset for supervised learning to teach an agent a mapping from states to actions, as in Behavior Cloning (BC), or the basis for learning a reward function, as in Inverse Reinforcement Learning (IRL). LfD is helpful when the intended behavior is difficult to design for with either control code or a designed reward function (Ravichandar et al., 2020). However, these methods have significant weaknesses. As a BC agent acts, its mistakes compound quadratically with the time-horizon (Ross et al., 2011) due to distributional shift, and IRL is an under-specified problem and may have difficulty generalizing from few demonstrations (Finn et al., 2016; Arora & Doshi, 2021). These difficulties arise in part from estimating human behavior



(a) PinPad5 (b) MemoryMaze 9x9

Figure 1: Training environments. (a) PinPad5: The agent must hit a specific 5-pad sequence in order to attain sparse reward. The agent's history accrues in the bottom left of the image (this agent has stepped on three pads). The pad sequence is the same for every trial (b) MemoryMaze 9x9: a 3D randomized maze environment where an agent receives reward for stepping on the correct hemisphere. The border of the observation indicates the target hemisphere color. Episodes last 1000 steps in both environments.

distributions. We avoid the problems of policy or reward shaping by providing demonstrations that train the world model, rather than an agent.

2.2 Learning from World Models

World models learn a representation of the environment's transition function $T : S \times a \rightarrow S', R, c$ that maps the current state S and an action $a \in A$ to the next state S', the environmental reward R, and (optionally) the likelihood that the episode will terminate c. World models allow RL agents to generate synthetic rollouts to learn from a much larger more diverse set of experiences than might be feasible in the real environment, increasing sample efficiency (Moerland et al., 2023). In addition, MBRL can aid exploration by using representation loss as a proxy for state-transition familiarity, and explainability by providing visualizable examples of future behavior. Model-based systems have become more common as high fidelity world modelling has improved (Ha & Schmidhuber, 2018; Kaiser et al., 2019; Hafner et al., 2019; Rafailov et al., 2021; Wu et al., 2023).

We base our system off of the Dreamer line of work (Hafner et al., 2019; 2022; 2023). Dreamer uses a Recurrent State Space Machine (RSSM) which encodes observations x, and joins them with deterministic recurrent state h to predict a stochastic z from past actions and embeddings. Dreamer learns to remember salient features over multiple time-steps, but outputs Markovian states which facilitate learning from reward signals. Dreamer trains an actor-critic agent that learns purely from imagined world model trajectories, which are generated from previously observed states. We insert human-interaction periods into the training-evaluation loop and let a human teleoperate an agent towards a sparse reward. See appendix A for more detail on Dreamer's world model and training.

2.3 LfD with Few Demonstrations

Recent years have seen progress in learning from small amounts of human demonstration data. These methods demonstrate how effective small amounts of human demonstration can be on forming useful embedded representations for model-free learning (Zhan et al., 2021), jointly training a world model and policy quickly (Hansen et al., 2022a), and training adversarial discriminators from world model

Algorithm 1 WMHD-Dreamer

```
Require: $W, \pi_{human}, \pi_{dream}, \pi_{random}, D$
$n_{human} \leftarrow 0$
$D \leftarrow s, \pi_{random}(s), s', r$ for $L$ steps
$D \leftarrow s, \pi_{human}(s), s', r$ for $L$ steps
for $i = 1 \ldots 100$ do
    train $W, \pi_{dream}$ on $D$
end for
while learning do
    for $j = 1 \ldots L$ do
        $D \leftarrow s, \pi_{dream}(s), s', r$
        train $W, \pi_{dream}$ on $D$
    end for
    if $n_{human} < n_{experiment\_samples}$ then
        $D \leftarrow s, \pi_{human}(s), s', r$ for $L$ steps
        $n_{human} \leftarrow n_{human} + L$
    end if
end while
```

unrolls (Rafailov et al., 2021). These methods are effectively used for continuous control tasks, but we push this research further by demonstrating its effectiveness on compositional tasks. Humans generally know the steps involved in accomplishing a task, but often struggle with directly controlling low-level agent behavior (Akgun et al., 2012).

3 Methodology

Our approach, World Model training from Human Demonstrations (WMHD), is based on DreamerV3 Hafner et al. (2023), which is composed of a world model consisting of an RSSM plus decoder heads for expected reward, episode termination, and image observation, and an actor-critic agent. The world model is trained on images from the environment, discrete actions and observed reward, while the agent is trained on forward predicted world model unrolls starting from real world states sampled from the replay buffer. See appendix A for more detail. We will distinguish our approach from pure Dreamer as *WMHD-Dreamer* in this text.

In baseline Dreamer training, a random policy generates trajectories, the system pretrains on that random data, and then the world model and agent are trained jointly to solve the task. WMHD-Dreamer speeds up learning by adding human demonstrations periodically early in learning using direct control / teleoperation. First, the expert human teleoperates the agent for one episode, then the system pretrains for 100 steps on a uniformly sampled mix of trajectories from human data and data produced by a random policy. The system then oscillates between dreamer agent control with training steps, and human control without training steps. Once the target number of human demonstrations have been collected ($n_{experiment\_samples}$), the agent proceeds with training without human intervention.

Algorithm 1 shows the training process with world model W, actor $\pi_{dreamer}$, random policy π_{random}, human π_{human}, and dataset D, for episodes of length L. $n_{experiment\_samples}$ was set to 1000, 3000, or 6000, in order to evaluate the effect of varying amounts of data on performance (see figure 3).

The algorithm was validated using a PyTorch implementation of DreamerV3 based on NM512 (2024)'s implementation, modified to allow for human demonstrations to be collected periodically within the training loop.

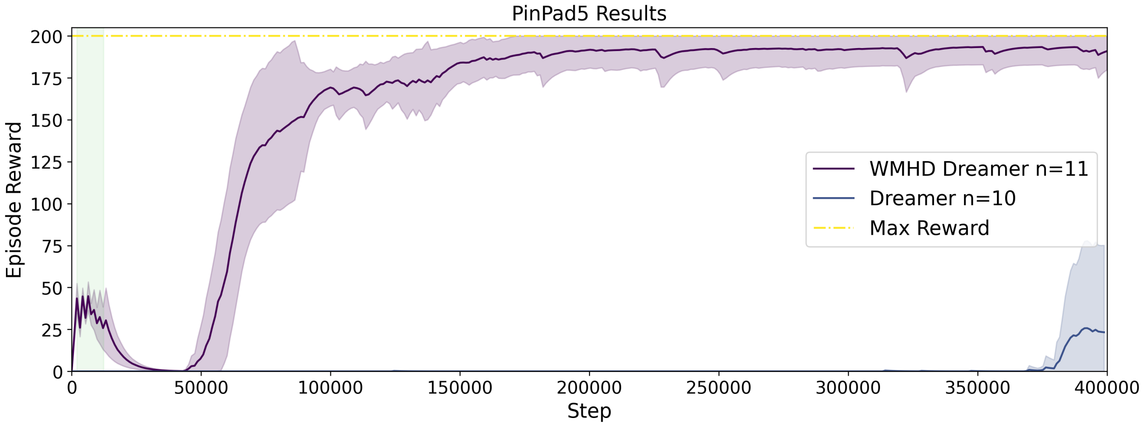


Figure 2: Episodic reward in the PinPad5 environment for 3k and 6k human actions demonstrated in the early stages of training vs. no human demonstrations. The shaded vertical region indicates when human demonstrations occurred. Lines indicate the mean across all trials with one standard deviation shaded (capped at 0 and max-reward). PPO, RS-PPO, and BC baselines never achieved any reward, and so have been omitted. Demonstrations were taken for each of n trials.

4 Simulation Study

We demonstrate the benefit of incorporating human data into world model training in two simulated environments. The first is a long-horizon task, PinPad5, where a pixel agent has to step on five colored pads in the correct sequence. The agent starts in a random location and gets credit for pressing a pad when they move onto any square of that pad, unless it was the last pad visited (a pad cannot be activated twice in a row). The pad-visitation history is tracked on the bottom left of the image observation (figure 1a), making PinPad5 a fully observable environment. PinPad5 takes discrete actions and returns 64x64 image observations, and has a relatively simple transition functions that is easy to control, but requires long-horizon planning and precise execution to find the sparse environment reward. PinPad5 is a *compositional* rather than a *control* task, because the task's challenge comes from visiting a long sequence of correctly ordered states rather than maintaining continuous control of an agent.

We also evaluate our method in MemoryMaze 9x9 (Pasukonis et al., 2022), a 3D randomized maze environment where an agent receives reward for stepping on the correct colored hemisphere. The target hemisphere color is indicated by the border of the 64x64 image observation, and is randomly selected once the previous target hemisphere is reached. This environment uses discrete-actions and presents image observations from a first-person perspective, making it a partially observable environment. It is designed to test long-term memory and exploration. In this environment, we compare our approach to Dreamer and PPO baselines.

All human demonstrations are collected from human experts (authors) using a wired Xbox controller. The expert is shown the environment's 64x64 image state and uses the joystick to input a direction that maps to a discrete environmental action. This process repeats for one thousand s, a, s', r transitions and takes between two and three minutes. The Dreamer agent then acts one thousand environment steps (125 updates) and hands control back to the human. This process continues until the target number of human transitions is collected. Each sample (seed) was trained for at least four hundred thousand steps corresponding to fifty thousand updates on either an Intel i7-12700 CPU with a GeForce RTX 3060 GPU or an Intel i5-10600K CPU with a GeForce RTX 3080ti GPU where Dreamer ran at approximately ninety-two updates per minute in PinPad5 and approximately fifty-five updates per minute in MemoryMaze9x9. Hyperparameters were kept constant across all experiments and can be found in appendix B. WMHD-Dreamer was trained with between nine and eighteen minutes of human demonstration, corresponding to three thousand to six thousand actions.

Algorithm	Steps to 50/90 %Max R	Trials 50/90 %Max R	Avg/StDev R
PinPad5			
WMHD-Dreamer	**6.0e4/1.1e5**	**100%/73%**	**162.74 / 66.06**
Dreamer	4.0e5/(n/a)	30%/0%	15.78 / 50.63
PPO, RS-PPO, BC	n/a	0%	0 / n/a
MemoryMaze			
WMHD-Dreamer	**4.8e4 / 6.1e4**	**100%**	**5.86 / 4.58**
Dreamer	2.0e5 / 2.7e5	**100%**	2.53 / 2.89
PPO	n/a	0%	0.15 / 0.12
RS-PPO	n/a	0%	0.14 / 0.12
BC	n/a	0%	0 / n/a

Table 1: Results over 400k Environment Steps. The first column shows the average number of steps an algorithm took to achieve either 50% or 90% of the maximum episodic reward. The second column shows what percent of trials (seeds) achieved 50% or 90% of maximum reward within 400k environment steps. The third column shows the average and standard deviation of the environmental reward for all steps and trials. For MemoryMaze, we use the maximum reward any baseline attained, which is roughly 25% of the reported mean maximum score after 100 million environment steps (Pasukonis et al., 2022).

We select our training window to overlap with an eight-hour workday in order to demonstrate learning at human timescales. Eight hours corresponds to 350k environment steps in PinPad5 and 225k environment steps in MemoryMaze on the consumer GPUs listed above. After testing with the Dreamer baseline, this window was extended to 400k steps to include the point at which dreamer starts to learn.

4.1 Baselines

We trained three baselines for PinPad5: PPO (Schulman et al., 2017b), PPO with a shaped reward (RS-PPO), and BC with the same demonstrations that were used to train WMHD-Dreamer. PPO with shaped reward received +0.2 for a correct pad in PinPad5 and a reward based on distance from the target hemisphere in MemoryMaze. For BC baselines, we trained with cross-entropy loss using a supervised learning model (see appendix 4) from 6000 s, a human demonstrations to predict the correct action for a given state and then evaluated the model's performance on the task. For hyperparameter choices, see appendix B.

In PinPad5, none of these agents achieved the sparse reward over the training horizon of 400000 steps and so are omitted from figures 2, 3. This is consistent with prior results: (Hafner et al., 2023) trained PPO for 30 million environment interactions without finding sparse reward in PinPad5.

5 Results

WMHD-Dreamer learned significantly faster and more consistently than Dreamer or any baseline. The results are strongest for PinPad5 (figure 2). In this environment a relatively small amount of human guidance leads to attaining 50% max environmental reward in 6.62 times fewer steps on average than pure Dreamer (60463 vs. 400350), and attains 90% max reward in 109519 on average while no baseline reaches 90% max reward. All ten WMHD-Dreamer trials achieve 50% of max reward, while only three of ten plain Dreamer trials do. In addition, eight of eleven WMHD-Dreamer trials attained 90% of the max reward, while plain Dreamer never reached 90% max reward within the training window. Our PPO and BC baselines never reached the sparse reward. MemoryMaze performed similarly (figure 4) attaining 50% of demonstrator's max reward[1] in 4.17 times fewer steps

[1] We use the maximum reward attained by any baseline, which was WMHD-Dreamer and reaches roughly 25% of the reported mean maximum score after 100 million environment steps (Pasukonis et al., 2022).

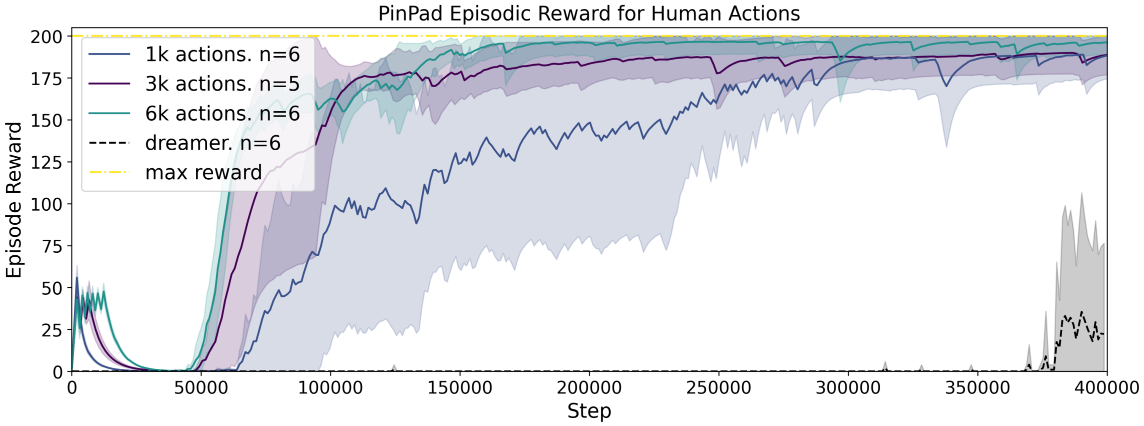


Figure 3: Episodic reward in the PinPad5 environment for differing amounts of human demonstration vs. no human demonstrations. Lines indicate the mean across all trials with one standard deviation shaded (capped at 0 and max-reward). Demonstrations were taken for each of n trials.

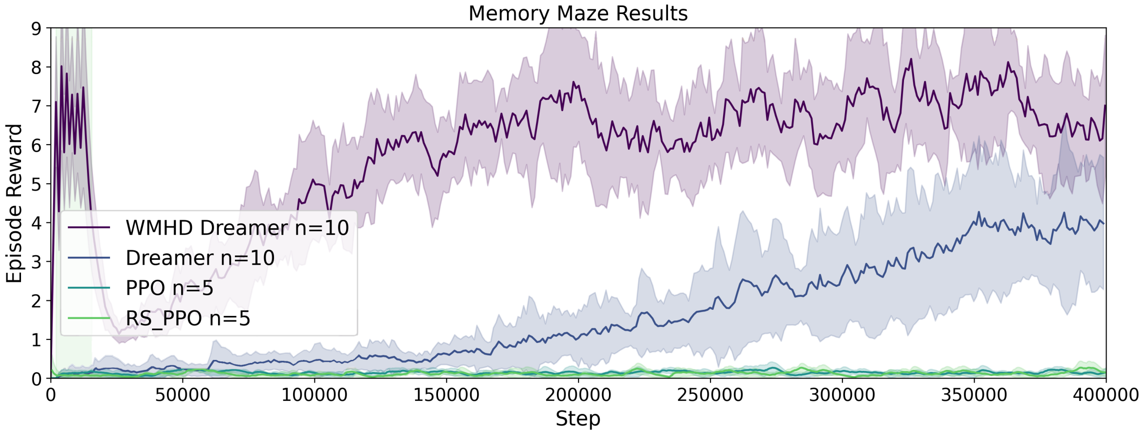


Figure 4: Episodic reward in the MemoryMaze9x9 environment for 3k and 6k human actions demonstrated in the early stages of training vs. no human demonstrations. The shaded vertical region starting at 0 steps indicates when human demonstrations occurred. Lines indicate the mean across all trials with one standard deviation shaded (capped at 0). Demonstrations were taken for each of n trials.

on average (48887 vs. 203759), and attaining 90% demonstrator's max reward in 4.00 times fewer steps on average (67405 vs. 269395). In this environment, all dreamer-based trials achieved 90% of demonstrator's max reward within the training window, while our PPO and BC baselines did not.

Figure 3 separates out PinPad5 trials by the amount of human demonstration data. We show results for between three and eighteen minutes, corresponding to between 1k and 6k human actions. Any human demonstration improves performance rapidly above pure Dreamer, but 3k and 6k demonstrations show much tighter variance and consistent performance than 1k demonstrations. Models with 6k demonstrations had an average reward of 169.83 and standard deviation of 57.38, 3k had an average reward of 172.05 and standard deviation of 50.60, and 1k had an average reward of 140.47 and standard deviation of 76.11.

6 Discussion

In this paper, we show that providing human demonstrations early in world model training results in significant speed-up and reliability improvements for two sparse reward tasks. In PinPad5, Dreamer starts learning around 400k steps which corresponds to nine hours on consumer hardware, while WMHD-Dreamer begins to learn at around 60k steps which is less than two hours of training. By introducing nine to eighteen minutes of human interaction, seven hours of training time are saved. Similarly, in MemoryMaze WMHD-Dreamer makes significant learning progress in the first 75k environment steps (less than three hours), while pure Dreamer needs 400k steps (sixteen hours) to approach the same levels of performance.

MemoryMaze performance exhibits more variance within the training window than PinPad5. This is partially due to the more random nature of the environment (e.g. some maze configurations provide easy access to all the hemispheres). In addition, MemoryMaze is an exploration and memory task where reward is attainable with less long-term precision than is required for PinPad5, so the reduced performance is in line with our expectation that the world model will benefit more from human in strongly compositional tasks. MemoryMaze still sees a 400% speedup within the training window because the human demonstrations contain important information for attaining sparse reward, e.g. that the border of the observation is the same color as the target hemisphere.

There are several possible explanations for WMHD-Dreamer's strong results. First, in sparse reward tasks, human demonstrations produce trajectories that are more spread out across the observation space than the world model would otherwise experience until much later in training. Each state on these trajectories becomes the initial state of an imagined trajectory for the actor-critic's on-policy learning. As a result, most training on sampled human demonstration is done *along trajectories that lead to sparse rewards* (see A.2 for a graphical depiction). This biases the world model representations towards accurately representing promising sub-trajectories, and forces dense exploration along, fruitful trajectories.

Training a world model in this way is also resilient to imperfect human demonstrations. Other policy shaping methods incur a temporal penalty for pushing an agent's policy towards useless or counterproductive human behavior, but our effect on the agent's policy is *indirect*, and occurs through the world model. Any useful information present in the trajectories can be incorporated into the world model, and this is especially potent because the alternative to human data is data produced by a mostly untrained or random policy. As long as the demonstrator is able to drive the agent to its goal, it should benefit the world model's understanding of the task.

6.1 Limitations and Future Work

This work examined the effect of human demonstrations on one world model. It is possible that the Dreamer architecture is particularly well suited to absorb information from a limited number of human demonstrations, and we can not claim this effect would persist if a different world model was used. The generalizability of this approach should be tested with other performant world models, like those used in TD-MPC (Hansen et al., 2022b) and VMAIL (Rafailov et al., 2021). In addition we did not test our system with continuous real-time control tasks, a common domain for MBRL, due to constraints on human response times. Follow up work could use assistive control to address these domains.

Also, this work does not examine how performance could be further improved through the use of interactive imitation learning. This paper claims WMHD-Dreamer is successful because it does not attempt to estimate behavior distributions, but it does not investigate whether policy shaping would further improve WMHD-Dreamer's performance. Future work should investigate the effect of other IIL techniques, like preference learning or scalar feedback, on MBRL systems.

Our primary insight is that training a state-of-the-art world model rather than directly shaping a policy results in a substantial speed-up. This insight may have benefits for IIL more generally and could be used incorporate human-demonstrations that understand harmful, or preferred states as well, offering a novel way to ensure safe or customized learned behaviors. In addition, a human observing a system in operation may notice behavioral weaknesses and take over to guide the agent through difficult, dangerous, or just sparsely explored sections of the state-space.

7 Conclusion

We demonstrate that a small number of human demonstrations can be leveraged by state-of-the-art world-model based reinforcement learning systems to dramatically decrease learning time and improve learning consistency. Our approach avoids the pitfalls of policy shaping by using human demonstrations to influence the world model rather than the acting agent. We show this effect in a sparse reward compositional task where we see six times faster, more consistent learning, and in a continuous space, discrete action memory and exploration task where we see more modest, but still significant learning improvements. This insight should be leveraged with modern interactive imitation learning methods to expand the effect of human-in-the-loop learning.

References

Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 391–398, 2012.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, Abhinav Valada, Jens Kober, et al. Interactive imitation learning in robotics: A survey. *Foundations and Trends® in Robotics*, 10(1-2):1–197, 2022.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pp. 49–58. PMLR, 2016.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 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*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

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

Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. *arXiv preprint arXiv:2212.05698*, 2022a.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022b.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019.

W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

NM512. dreamerv3-torch: A PyTorch implementation of DreamerV3. https://github.com/NM512/dreamerv3-torch, 2024. Accessed: 2024-01-04.

Jurgis Pasukonis, Timothy Lillicrap, and Danijar Hafner. Evaluating long-term memory in 3d mazes. *arXiv preprint arXiv:2210.13383*, 2022.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models. *Advances in Neural Information Processing Systems*, 34:3016–3028, 2021.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017a. URL http://arxiv.org/abs/1707.06347.

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

Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions. In *Robotics: Science and Systems (RSS)*, 2020.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Marcel Torné Villasevil, Max Balsells I Pamies, Zihan Wang, Samedh Desai, Tao Chen, Pulkit Agrawal, and Abhishek Gupta. Breadcrumbs to the goal: Supervised goal selection from human in the loop feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

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

Albert Zhan, Ruihan Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. A framework for efficient robotic manipulation. In *Deep RL Workshop NeurIPS 2021*, 2021.

A Dreamer World Model

The world model learns embedded representations of the input through auto-encoding and recurrence. It is built from a PyTorch port (NM512, 2024) of Hafner et al. (2023) Recurrent State-Space Model (RSSM), which maps inputs obs_t to stochastic output z_t through a deterministic sequential model with hidden state h_t (Hafner et al., 2019; 2023). A Gated Recurrent Network (GRU) predicts the next deterministic state from the previous deterministic state, and an MLP combination of the previous action and the previous stochastic state 5. The GRU's prior is then MLP combined with the encoded current observation to obtain the deterministic posterior state of the world. The model learns a stable, long-term embedded world state in h, but can handle the stochastic nature of complex unobservable environments by updating from the stochastic state. Agents can train on both the deterministic and stochastic states to actualize in the real-world.

The world model can be represented by the following equations. Where h_t is the deterministic recurrent state, z_t is the embedded stochastic state, x_t is the encoded observation, $\hat{z}_t$ is the predicted stochastic state, $\hat{r}_t$ is the predicted reward, $\hat{c}_t$ is the predicted likelihood of the episode continuing, and $\hat{x}_t$ is the decoded image.

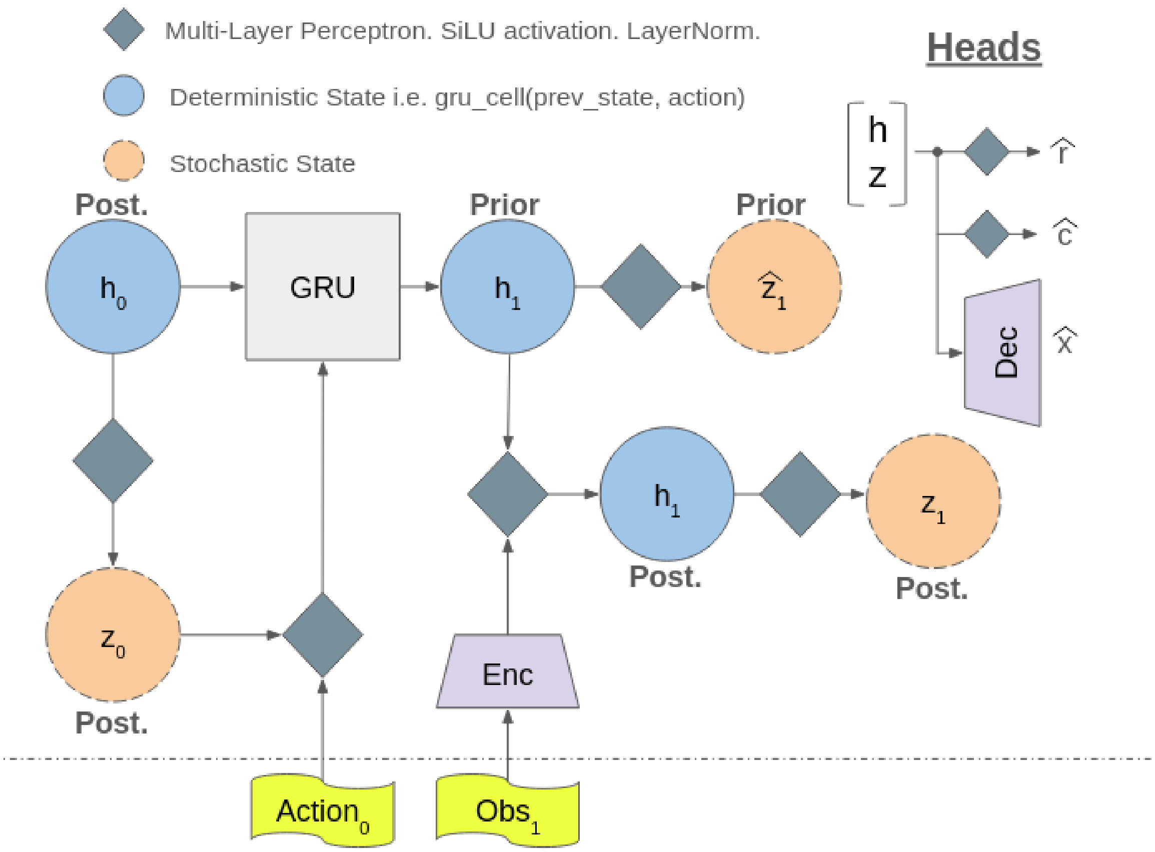
$$\begin{gathered}
h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\
z_t \sim q_\phi(z_t | h_t, x_t) \\
\hat{z}_t \sim p_\phi(\hat{z}_t | h_t) \\
\hat{r}_t \sim p_\phi(\hat{r}_t | h_t, z_t) \\
\hat{c}_t \sim p_\phi(\hat{c}_t | h_t, z_t) \\
\hat{x}_t \sim p_\phi(\hat{x}_t | h_t, z_t)
\end{gathered}$$

A.1 RSSM structure

Figure 5 shows the one-step update for the RSSM. See Hafner et al. (2023) for more details.

A.2 Imagined Future Trajectory Training

Dreamer trains actor-critic agents on imagined trajectory unrolls that start from each real state-action pair it observes. A single actor-critic is trained on extrinsic (usually environmental) reward as well as an entropy regularizing term to encourage exploration. Figure 6 shows an example training batch from an expert demonstration. Expert states S_t^E form the basis of imagined unrolls where π learns on-policy. Expert actions a_t^E are never considered in actor-critic training.



Multi-Layer Perceptron. SiLU activation. LayerNorm.
Deterministic State i.e. gru_cell(prev_state, action)
Stochastic State
Heads
Post.
h0
GRU
Prior
h1
Prior
ẑ1
h
z
r̂
ĉ
Dec
x̂
h1
z1
Post.
Post.
z0
Post.
Enc
Action0
Obs1

Figure 5: One-step RSSM with image observation encoder and reward, likelihood of continuation, and observation decoders.

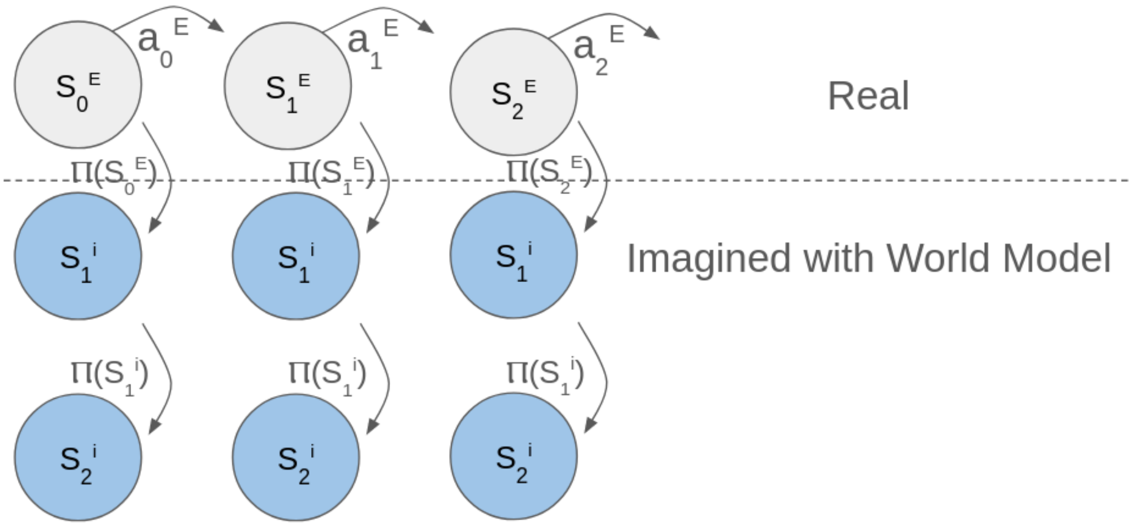


Figure 6: One-step RSSM with image observation encoder and reward, likelihood of continuation, and observation decoders.

B Hyperparameters

Table 2 shows the hyperparameters used to train WMHD-Dreamer and Dreamer.

Hyperparameter	**Value**
World Learning Rate	1e-4
Actor Learning Rate	3e-4
Critic Learning Rate	3e-4
Train Ratio	128
GRU Recurrent Units	1024
CNN Multiplier	32
Dense Hidden Units	512
MLP Layers	4
Human Interaction Period (env steps)	1000

Table 2: Dreamer hyper-parameters

Hyperparameter	**Value**
Actor Learning Rate	3e-4
Action Std Decay Rate	0.05
Min Action Std	0.1
Critic Learning Rate	1e-3
Gamma	0.99
Epsilon Clip	0.2

Table 3: PPO hyper-parameters

Table 3 shows the Proximal Policy Optimization Schulman et al. (2017a) hyperparameters used to train the baseline approach.

Hyperparameter	**Value**
discount factor	0.995
learning rate	1×10^{-3}
optimizer	Adam
batch size	256
action distribution	categorical with 4 bins
model architecture	2 Conv layers followed by 2 linear layers

Table 4: Behavior Cloning Model hyper-parameters

Table 4 shows the Behavior Cloning hyperparameters used to train the baseline approach. The behavior cloning model was trained using 6000 human demonstrations, and the model achieved a test accuracy of 65%. The train-validation-test split was 80-10-10.