Imitation learning through imagination in latent space

Ekaterina Amozova
School of Computing
University of Eastern Finland
Joensuu, Finland
ekaterina.amozova@uef.fi

Janne Laakkonen
School of Computing
University of Eastern Finland
Joensuu, Finland
janne.laakkonen@uef.fi

Federico Malato
Department of Pediatrics
University of Iowa
Iowa City, Iowa
federico-malato@uiowa.edu

Ville Hautamäki
School of Computing
University of Eastern Finland
Joensuu, Finland
ville.hautamaki@uef.fi

Abstract

Learning a policy to reach vague goal states that cannot be specified mathematically is a daunting task. A recent example from the NeurIPS BASALT challenge is to build a house in Minecraft. Formulating a reward function for such a task is impossible. Approximating a reward model based on a set of expert demonstrations via e.g. adversarial imitation learning eases the task by reducing it to a distribution matching problem. However, adversarial training is inherently unstable and imposes a significant computational cost due to the high-dimensional space in which it operates. In this work, we leverage advancements from recent models like DreamerV3 to explore the estimation of a reward model directly in latent space, potentially reducing the computational demands of adversarial training. We show that through comparing imagination rollouts to expert gameplay in latent space, the agent learns a meaningful policy even with limited amount of expert data.

1. Introduction

Many real-world decision-making tasks lack a clearly defined reward signal, which makes *reinforcement learning* (RL) [41] difficult to apply effectively. Reward shaping can go wrong in several ways: overlooking seemingly irrelevant aspects of the environment can lead to harmful side effects; alternatively, the designed goal might lead an agent to exploit an environment through undesired strategies [1]. In general, encompassing all possible scenarios coherently with a single reward function is not feasible. For complex tasks, where specifying a reward signal may be hard or even impossible, *imitation learning* (IL) [38] is generally preferred to standard RL. IL is a natural choice when aiming to train an agent to leverage human experience without an explicitly defined reward function: an IL agent learns by first observing an expert completing a task and then mimicking it.

Behavioral cloning (BC) [3, 37] is a method that turns the IL problem into a supervised learning classification problem, directly mapping states to actions based on expert demonstrations [7]. Well-known flaws of this approach include distributional shift [36] and poor generalization, resulting in the need to use large amounts of data for training. *Inverse reinforcement learning* (IRL) [31] can be used to infer a reward function from expert data, from which a policy is then extracted via RL. Despite some success in control tasks [31], IRL has not been successfully applied to larger problems due to its high computational cost. *Generative adversarial imitation learning* (GAIL) [22] was proposed as an alternative solution to this problem. GAIL addresses the problems of IRL by jointly learning a policy and a reward model in an adversarial fashion. In general, GAIL requires a small amount of expert data to recover a good policy, but its training is slow and unstable, like other adversarial methods [2, 25].

World models [16] can be used to explicitly learn the underlying dynamics of an environment, allowing agents to make informed predictions about the future and improving generalization. An interesting concept in this domain is imagination training - learning from hypothetical experiences generated by the world model [40], which improves sample efficiency. To contain computational costs while preserving information, latent dynamics models were shown to capture the necessary elements of the environ-

ment and allow for accurate long-term planning [15, 43]. An example of a model-based algorithm that benefits from both imagination training and long-term predictions in latent space is Dreamer [19–21].

Dreamer is a novel and efficient algorithm that leverages past experiences to learn a world model and imagined trajectories to learn long-horizon behaviors. Its architecture consists of three modules: a dynamics model (world model), a reward model (critic), and a policy (actor). The algorithm trains all components concurrently. More specifically, Dreamer updates its policy using imagined experience; then, it uses the improved policy to collect new rollouts in the real environment, and uses this experience to update the world and reward models.

As such, Dreamer's reward model depends on the actual rewards coming from the environment. Therefore, Dreamer is not applicable for tasks that do not feature a reward signal. In general, specifying a reward signal for RL is a hard problem [29]. Additionally, in some cases a reward might not be available at all. This is especially true for complex environments [28, 29].

To address this shortcoming, we take inspiration from adversarial imitation learning. The idea of combining the advantages of adversarial imitation learning with Dreamer-like, model-based algorithms has been explored previously. Our work builds on top of V-MAIL [34], an algorithm that applies GAIL to a Dreamer-like model using a variational latent-space dynamics model. The authors highlight V-MAIL sample efficiency and learning stability. However, V-MAIL was only applied to simple control tasks. Hence, its effectiveness on more complex domains, like open-world games, has yet to be investigated.

In this paper, we address the dependency of DreamerV3 on pre-defined reward signals. Taking inspiration from V-MAIL, we apply adversarial imitation learning directly to DreamerV3. We end up obtaining close to PPO level performance without access to reward signal and with a limited dataset of expert trajectories. Additionally, we explore the applicability of V-MAIL-like approaches to more complex tasks, such as open-world games.

We summarize our contributions as:

- We introduce **Dreamer-GAIL**, a reward-free imitation learning framework that integrates adversarial imitation directly into **DreamerV3**'s latent imagination space, enabling policy learning without access to environment rewards.
- We propose an **imitation-through-imagination** paradigm that adversarially matches imagined latent rollouts to expert trajectories, and provide a **theoretical analysis** showing improved generalization and bounded deviation from classic GAIL under approximate encoder sufficiency and bounded world-model error.
- We empirically demonstrate that the proposed method **outperforms standard imitation learning baselines** and demonstrates strong performance relative to PPO on Atari and Crafter using a **limited number of expert demonstrations**.

2. Related Works

World models: Classic RL policies [27, 39, 41] have been proven to be unfeasible for complex tasks due to their inherent sample inefficiency. World models [16, 24] have addressed this problem by letting a policy train on imagined states. Furthermore, Hafner et al. [19] introduced the idea of a recurrent simulator [8] to train a policy in a simulated environment. Later versions of the architecture—described by Hafner et al. [20] and Hafner et al. [21]—improved the representation by using categorical random variables and extended the applicability to a variety of domains, respectively. The Dreamer model has also been enhanced by other teams, for example, by including diversity in imagined trajectories to improve sample efficiency and model robustness [30].

For Dreamer, it is easy to lose small details in the decoded outputs, a fact alleviated by categorical latent space [20], but the true culprit appears to be the log-likelihood reconstruction loss [32]. Removing the decoder completely is a reasonable solution, but will still require a learning signal to estimate the encoder. The solution proposed by Okada and Taniguchi [32] is to utilize a discriminator. There is an analogy to our proposed method, as we replace the original reward model with a discriminator.

Imitation Learning: As another attempt to improve the sample inefficiency of RL, IL paved the way for the introduction of offline reinforcement learning [26]. BC [3, 37] uses the supervised learning paradigm to predict actions from observations. Drawing inspiration from Generative Adversarial Networks (GANs) [14], methods such as Adversarial Inverse Reinforcement Learning (AIRL) [11] and Generative Adversarial Imitation Learning (GAIL) [22] introduced the concept of adversarial training for IL and IRL. Building on these, MGAIL [4] combined adversarial training with a forward model that learns the dynamics of an environment, while V-MAIL [34] extended this approach to high-dimensional tasks.

The goal of imitation learning can be thought of as finding a policy that will produce a marginal distribution of rollouts that is close to the marginal distribution of expert demonstrations. One way to accomplish this is via Primal Wasserstein Imitation Learning (PWIL) [10, 42], where a policy is learned via a standard reinforcement learning algorithm, but the reward is based on closeness to the visited states.

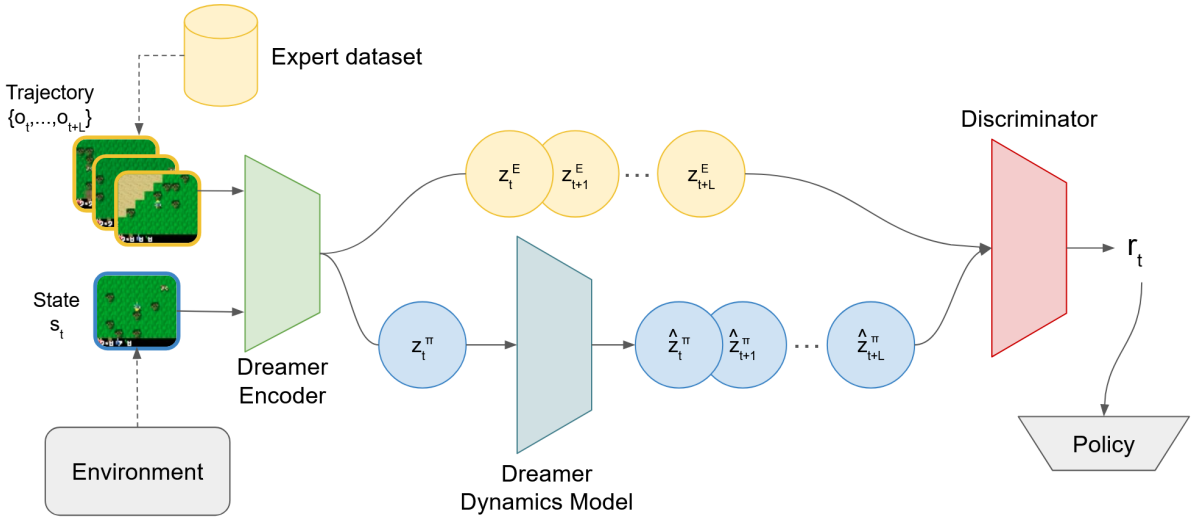


Figure 1. An overview of our method. At timestep t, the environment provides a state s_t that is passed through Dreamer. As a result, an imagined sequence of length L is obtained. Concurrently, a trajectory of length L is sampled from the expert dataset and each observation is encoded using Dreamer encoder module. Both expert and imagined trajectories are then fed to our discriminator to compute a reward signal r_t. Finally, the inferred reward is used to update the current policy π.

3. Problem Formulation and Method

3.1. Markov Decision Process

We assume that our reinforcement learning problem follows the *Markov decision process* (MDP) [6, 41] paradigm. An MDP can be completely described by a 4-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, where $\mathcal{S}$ is the set of allowed states an agent can visit and $\mathcal{A}$ is the set of possible actions; $\mathcal{T}$, known as the *transition probability*, describes the probability of visiting state s' from the current state s when executing action a, i.e. $\mathcal{T}(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$; finally, the *reward function* $\mathcal{R}(s,a) \to \mathbb{R}$ provides a scalar feedback to an agent visiting state s and executing action a. In the RL setting, the main learning goal is to find a policy $\pi : \mathcal{S} \to \text{Prob}(\mathcal{A})$ that maximizes the discounted cumulative reward $\mathbb{E}[\sum_{t=\tau}^{T} \gamma^{t-\tau} r(s_t, a_t)]$, with T being the length of an episode, $r(s_t, a_t)$ the realized reward at timestep t, and $\gamma \in [0,1]$ is a *discount factor*. Notably, policies are agnostic to the transition dynamics $\mathcal{T}$.

The imitation learning scenario follows the same MDP paradigm. However, in this case the reward function is also unknown. In IL, our goal is to extract a policy π that solves the task at hand by leveraging a pre-collected set of expert demonstrations. In general, demonstrations are encoded as *trajectories* of state-action pairs $\tau_i = \{(s_0, a_0), (s_1, a_1), \ldots, (s_{T-1}, a_{T-1})\}$ stored in a dataset $\mathcal{D} = \{\tau_1, \tau_2, \ldots, \tau_N\}$, where N is the number of available trajectories. These key differences profoundly influence the learning paradigm: for instance, an IL agent may attempt to mimic the expert directly; on the other hand, some IL algorithms focus on inferring a reward signal from demonstrations and extracting a policy from it. In the latter example, the goal of IL can be stated as finding a combination of policy and reward model that minimizes the distance between the marginal distributions of the policy-generated *rollouts* and expert demonstrations.

3.2. Proposed Method

Dreamer consists of three neural networks: world model, actor, and critic. In our work, we preserve the world model while replacing the reward model with an adversarial discriminator. The discriminator learns to distinguish between the predictions produced by the combination of actor and dynamics model and the expert gameplay, simultaneously training the policy to generate trajectories that are closer to the experts' trajectories. Technically, our implementation has the discriminator as a part of the actor-critic ensemble. The reward head of Dreamer's world model is simply unused: we remap the reward function to the output of the discriminator. World model loss remains unaffected. Actor and critic losses are affected only in the sense that the critic now uses a custom reward function to score policy trajectories. The process is illustrated in Figure 1. In Algorithm 1 we can see the pseudo-code of the Dreamer-GAIL training loop. The adversarial part, which is our proposed addition, is highlighted by color. The model states $m_t \approx \{h_t, z_t\}$ mentioned in the pseudo-code encompass the deterministic state h_t produced by Dreamer's sequence model combined with the stochastic representation z_t. This notation is different from the one used by [21], so as not to confuse model states specific to Dreamer with generic states as in state-action pairs.

Algorithm 1 Dreamer with AIL

Initialize dataset B_π with random seed episodes
Randomly initialize neural network parameters for Dreamer components and discriminator D
for number of iterations **do**
  for update step **do**
    // Dynamics learning
    Sample B trajectories of length L from B_π
    Map inputs x_t to stochastic representations z_t and predict the sequence of representations given past actions
    From model states m_t, predict continuation flags c_t
    From model states m_t, calculate predicted rewards r_t using chosen reward function
    Reconstruct the inputs $\hat{x}_t$
    Update the world model
    // Adversarial policy learning
    Sample B trajectories of length L from expert dataset B_E
    Infer expert latent states z^E from expert observations x^E
    Generate latent rollouts (imagine trajectories) z^π from policy dataset B_π
    Update the discriminator D using$\{z^\mathrm{E}, a^\mathrm{E}\}$ and $\{z^\pi, a^\pi\}$
    Update the policy π using chosen reward function r_t
  end for
  // Environment interaction
  Reset environment
  for time step $t = 1..T$ **do**
    Compute model states from history
    Sample action a_t from the actor model
    Environment step
  end for
  Add experience to B_π
end for

In the original Dreamer, the goal was to train the model based on imagined trajectories, but the model had access to the environment reward. However, such a reward is not directly associated with an imagined state-action pair (s, a). The trained reward model was then used to *predict* the realized reward $r(s, a)$ for the *imagined* state s. In the case of imitation learning, we need to train a discriminator $D(s, a)$ that takes imagined states and world model encoded expert demonstration states.

Orsini et al. [33] analyzed aspects of adversarial imitation learning in the observation domain. They list four possible reward functions as the original GAIL [22], AIRL [11], their own "Natural" and FAIRL [12] reward functions. In this work, based on the findings in [33], we use the GAIL reward function

$$r(s, a) = -\ln(1 - D(s, a)) \tag{1}$$

for our experiments. We have also explored the AIRL reward function, but the differences in performance were substantial.

3.3. Imitation Objective

Let $J(\pi)$ be true task return (environment score). Ultimately, we want to show that after N environment steps:

$$J(\pi_{DG}) \geq J(\pi_G) - (\text{error terms}), \tag{2}$$

where π_G is the policy learned by standard GAIL algorithm and π_{DG} is the policy learned by Dreamer-GAIL. To this end, we decompose the performance gap into the three error terms outlined below.

The change from raw image input to latent space inevitably introduces a representation error. Although latent representations are less noisy than high-dimensional input, they may also fail to capture relevant information for the task. To account for representation error, we assume that the encoder is approximately sufficient, in other words, we assume it preserves the information that the discriminator needs to distinguish between expert and agent behavior (see details in 10).

As Dreamer uses its learned dynamics model to produce imagined rollouts, they may differ from the actual environment experience, even when produced by the same policy. This, in turn, introduces a model error, which grows with imagination horizon.

4. Experimental Setup

Dreamer provides several pre-defined model sizes: small, medium, large and extra large. Size is directly proportional to both performance and data-efficiency, as it determines the complexity of the underlying recurrent state-space model (RSSM) [18]. In our experiments, we use a small size Dreamer model due to computational limitations. The hyperparameters are consistent with the reference model used in Hafner et al. [21]. The discriminator is implemented as an MLP with 2 hidden layers. The default learning rate of the Dreamer actor is 3e-5. According to Orsini et al. [33], it may be better for the discriminator to have a learning rate that is $2-2.5$ orders of magnitude lower than that of the RL agent, but we used a learning rate of 3e-9, as the discriminator was prone to learning too fast.

100M was chosen as the minimal number of environment steps to determine the performance of a model in the Atari games. For DreamerV3, 100M is the number of steps needed for the performance to start to plateau for most environments [21], so we assume that imitation learning would require at least as many interactions with the environment

to show meaningful progress or lack thereof. In Crafter, we have to settle for 20M steps because of compute limitations.

4.1. Baselines

RL baselines To benchmark our imitation learning approach, we include two model-free reinforcement learning baselines: Proximal Policy Optimization (PPO) [39] and Categorical DQN (C51) [5]. Both implementations are taken from the CleanRL v2.4 library [23].

All baselines share a common experimental setup. We use the standard Atari preprocessing pipeline: *MaxAndSkip(4)* $\rightarrow$ *grayscale resize* 84×84 $\rightarrow$ *frame-stack 4*. The agents use the canonical convolutional architecture and the default CleanRL hyper-parameters (e.g., Adam with a learning rate of 2.5×10^{-4}, $\gamma = 0.99$). Training is conducted for 50 million environment steps for each of five random seeds. During evaluation, we run the greedy policy for 10 episodes and report the mean and standard deviation of undiscounted returns.

PPO is run in an on-policy setting with 8 parallel environments; after collecting 128 transitions from each environment, the agent executes 4 epochs of updates on the aggregated batch using the clipped-surrogate objective, and learning-rate annealing remains enabled. C51, in contrast, is an off-policy agent that represents the return distribution with 51 atoms and learns from a 0.5 million transition replay buffer. Full experimental details are provided in 9.

IL baselines Additionally, we train two PPO agents using vanilla AIRL [11] and GAIL [22] on a small set of human-collected data. In our experiments, we use the implementation of the two algorithms provided in the *imitation* [13] library. To ensure a fair comparison with the RL baselines, we leave the hyper-parameters for PPO unchanged. Due to the instability issues of AIRL and GAIL, we ran a preliminary study to determine the best configuration of hyper-parameters for them. Unlike the reference implementation, we run both algorithms for 2.5M steps; additionally, during the adversarial loop, the discriminator is updated only once, to avoid extreme dominance over the generator. As the third IL baseline, we use MGAIL [4]. We modify the implementation to work with image input by replacing the encoder and decoder with their convolutional counterparts while leaving the structure unchanged.

4.2. Environments

The Atari environments have been widely used not only in RL research, but also for IL purposes [35, 44]; they provide a set of diverse challenges, and the visual demonstrations are easy to obtain. For our experiments, we use two Atari games: Seaquest and Alien. These environments are distinct enough to require different skills and provide a meaningful scale between the random agent score and the human score, as reported by Hafner et al. [21], making it easier to evaluate the performance of the algorithm. As a third environment, we use Crafter [17], a 2D open world survival game provided as one of the default environments available for Dreamer.

In Seaquest, the player is a submarine that rescues divers while avoiding or shooting enemies (sharks and other submarines). When the submarine is low on oxygen, it should come up to refill air at the surface, and the difficulty of the game will increase. Each time the submarine comes up without any divers on board, it will lose a life.

In Alien, the player should destroy alien eggs and avoid aliens. Destroying eggs should, in theory, discourage the player from visiting the same place twice, but moving enemies can complicate achieving this goal. Overall, the gameplay in Alien is less repetitive, the dynamics are more complex and theoretically harder to learn.

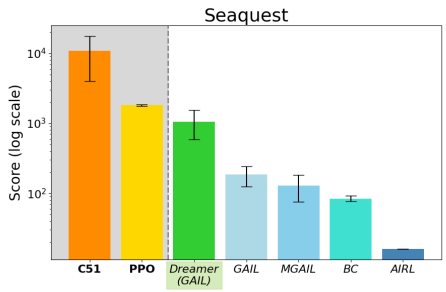
The Crafter environment features a hierarchical crafting system: some resources, such as stone, can be collected only upon crafting a specific tool; conversely, some tools can be crafted only after collecting a specific resource. Additionally, Crafter features enemies like zombies and skeletons that can harm the player. Due to this fast-paced game world, surviving in Crafter requires strong generalization abilities.

4.3. Dataset

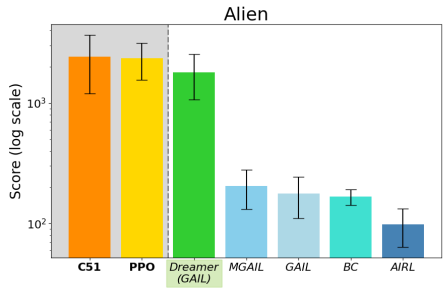
For both Seaquest and Alien, we construct our datasets from the trajectories included in the AtariHEAD dataset (Version 4, CC-BY 4.0) [46, 47]. We are only interested in RGB observations and actions. We only use the two best-score trajectories, assuming that the optimality of the expert policy correlates with the success of the resulting game. We slice each original trajectory into five parts featuring different stages of gameplay, thus obtaining 10 expert trajectories, each of length 1024. For Crafter, we use the human gameplay dataset provided by the author of the environment (Crafter Human Dataset, CC-BY 4.0) [17]. We only use the longest trajectories that tend to feature a wide variety of achievements and behaviors, obtaining 10 expert trajectories, each of length 1024.

4.4. Evaluation

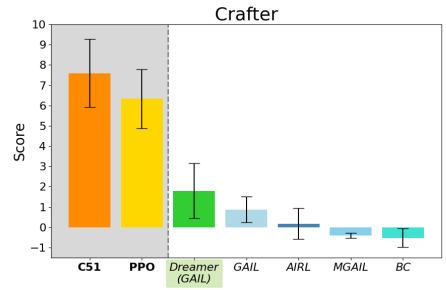
Performance is primarily determined by the in-game score. However, episode duration must also be considered: for instance, an agent could learn to early terminate an episode to avoid a negative score; alternatively, agents might learn to survive without accumulating score. Therefore, policies that produce either long, low-score or extremely short, non-negative-score episodes cannot be considered successful. For each environment, we provide a quantitative evaluation on a total of 250 episodes, aggregated over five chunks of 50 episodes played with different seeds.



(a) Evaluation scores for Seaquest.



(b) Evaluation scores for Alien.



(c) Evaluation scores for Crafter.

Figure 2. Evaluation scores. The shaded area denotes RL methods (C51, PPO), which are also labeled in bold. IL methods are labeled in italics. Our method (Dreamer-GAIL) is further highlighted by color. Note that Seaquest and Alien scores are reported using log scale, while Crafter score is reported as-is. The error bar is standard deviation.

Additionally, we qualitatively evaluate the agents on recorded episodes by inspecting consistent patterns in the learned behavior. We point out that, while this evaluation may not produce an accurate analysis of RL & IL algorithms, assessing complex behaviors through a scalar reward is generally difficult [29]. As such, we provide this additional analysis to highlight emerging strategies that might otherwise be overlooked.

5. Results

The comparison of Seaquest evaluation scores can be seen in Figure 2a. We notice that, as in the other two games, Dreamer-GAIL consistently ranks third, outperforming other IL baselines. We notice a performance gap of around 700 points between the proposed method, with GAIL reward, and the PPO. It is notable that under favorable seeds the proposed method almost reaches the performance level of the PPO and even at the unluckiest it still wins over the best baseline IL method, GAIL.

Our method was not intended to directly compete against the RL baselines, which were included to show the best performance if reward were available. In Figure 2b we see evaluation scores for Alien environment, and we notice that the proposed method is almost on par with RL baselines. Noting also that all baseline IL methods (GAIL, BC and AIRL) performed equally poorly. Despite using the same expert demonstrations as the IL baselines, the proposed method extracted more meaningful information from the data.

Finally, in Figure 2c we see evaluation scores of the Crafter environment. Again we notice the same trend, C51 is a bit better than PPO, and the proposed method is the winner over IL baselines. In the learning curves we see potential for improvement in Dreamer-GAIL, thus theoretically it could achieve the same in Crafter if trained for longer. However, the transition to open-world games highlights the need for a more diverse dataset, potentially including trajectories that are not successful in the strict sense but nevertheless provide additional examples of human behavior.

In Seaquest, Dreamer-GAIL has successfully learned to shoot or avoid enemies, rescue divers, and come to the surface when low on oxygen (Figure 3). In Alien, it has learned to avoid aliens long enough to destroy most of the eggs on the level (Figure 4). In Crafter, the agent has mostly learned to passively survive, but there was one particularly inventive seed that exhibited varied and successful behavior (Figure 5).

6. Connection to V-MAIL

As Dreamer-GAIL, also V-MAIL combines learning in latent space with a discriminator. Even though there are surface level similarities, significant differences also exist and these are explored in the following Section and rest of the discussion can be found in the Appendix F.

Latent space sufficiency. Let ϕ_{VAE} and ϕ_{DV3} denote the VAE encoder and DreamerV3 encoder respectively. Under the assumption that the VAE encoder is trained to minimize a reconstruction bound

$$\mathcal{L}_{VAE} = \mathbb{E}[-\log p(s|z)] + \mathrm{KL}[q(z|s)||p(z)], \quad (3)$$

the VAE objective optimizes for $\mathrm{I}(s; z)$, the mutual information between a state and its latent representation. By the data processing inequality, for any function f:

$$\mathrm{I}(f(s); z) \leq \mathrm{I}(s; z) \leq \mathrm{I}(s; s) - \mathrm{KL}[q(z|s)||p(z)]. \quad (4)$$

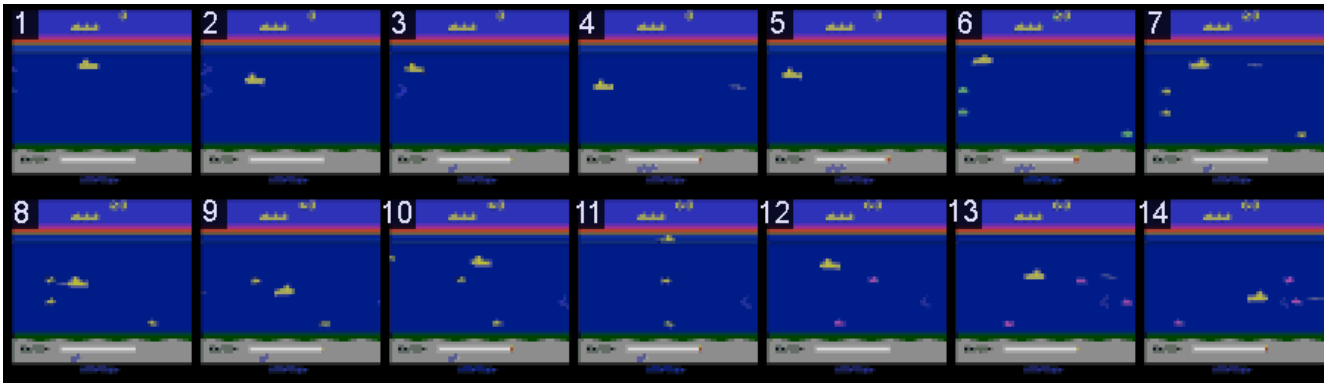


Figure 3. An example of successful behavior from Dreamer-GAIL in Seaquest, featuring each 8th frame. The agent purposefully collects divers (frames 3, 4) and brings them to the surface (frames 6, 11) while simultaneously shooting enemies. The offloading of divers on the surface raises the difficulty, which can be seen in the changing colors of the enemies.

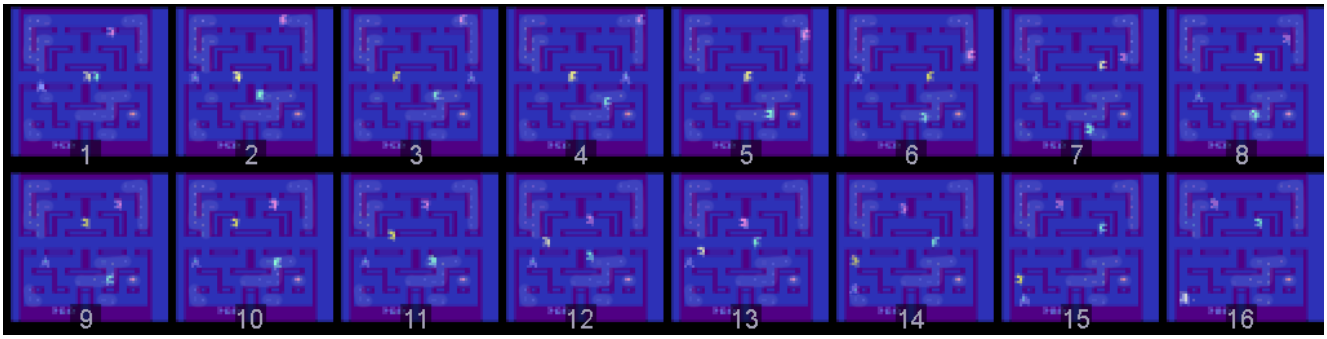


Figure 4. The ending of an episode for Dreamer-GAIL in Alien, featuring each 8th frame. The paths where alien eggs are intact are highlighted. While the agent hesitates in the frames 8-13, an alien approaching seems to motivate the agent to flee. The episode still terminates with the alien catching the agent.

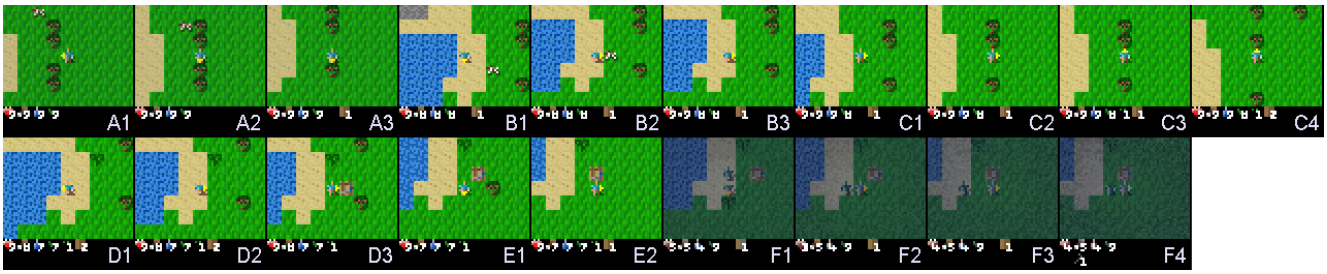


Figure 5. An example of successful behavior from Dreamer-GAIL in Crafter, all in the span of the same episode. The agent collects wood (frames A1-A3, C1-C4, E1-E2), uses 2 pieces of wood to construct a table (D1-D2) and 1 piece of wood to make a wooden pickaxe (F1-F4). In the same episode, the agent eats a moving cow (B1-B3).

The KL term strongly penalizes assigning low probability to real data, actively compressing the representation, and leading to potentially discarding relevant information. The DreamerV3 encoder, on the other hand, is tightly integrated with the dynamics predictor and is assumed to minimize a predictive bound

$$\mathcal{L}_{DV3} = \mathbb{E}[-\log p(s_{t+1}|z_t, a_t)], \tag{5}$$

which optimizes for $\mathrm{I}(z_t, a_t; z_{t+1})$, the mutual information between the current latent-action pair and the next latent state. Because the discriminator must distinguish behavioral trajectories, an optimized representation for transition prediction better preserves the information relevant to the discriminator.

Tailored for Continuous Actions. V-MAIL uses SAC as its underlying policy optimizer. The policy update (Equa-

tion 8 in the paper) back-propagates through the learned dynamics model $\hat{\mathcal{T}}_\theta$ to optimize the policy π_ψ.

This is the reparameterization trick applied through the model, which means: (i) The policy is assumed to be a continuous, differentiable distribution (a squashed Gaussian, as in standard SAC). (ii) The gradient path $\partial\mathcal{L}/\partial\pi_\psi$ through model rollouts requires differentiable action sampling. (iii) In a discrete setting, this gradient path breaks for the same reasons as in vanilla SAC.

Our experiments indicate that V-MAIl is able to learn to reconstruct states in the Atari environment (see Appendices for rollouts). But the learned policy is not able to take any actions. According to analysis above, we would need to apply discrete action space fixes developed for SAC such as in [9, 48].

7. Conclusions

This work demonstrates that imitation learning can be successfully performed without access to an explicit reward signal by leveraging latent imagination within a world-model framework. By integrating adversarial imitation learning into DreamerV3, we enable policies to be learned through comparisons between imagined trajectories and expert demonstrations in latent space. Our theoretical analysis provides insight into why this formulation improves discriminator generalization and stabilizes training relative to observation-space adversarial approaches. Empirically, we show that the proposed method consistently outperforms standard imitation learning baselines and narrows the gap to reward-driven reinforcement learning across Atari and Crafter, despite relying on a limited set of expert trajectories.

At the same time, our results highlight several limitations. Performance in open-world environments such as Crafter plateaus early, suggesting sensitivity to demonstration diversity and hierarchical environment dynamics. In addition, the approach inherits approximation error from both the learned world model and the latent representation, which may limit scalability to longer horizons or highly stochastic domains. Addressing these challenges through richer and more diverse expert datasets, improved world-model fidelity, and mechanisms for handling heterogeneous or suboptimal demonstrations represents an important direction for future work. Developing discrete action space fixes for V-MAIL would enable fair empirical comparison between Dreamer-GAIL and V-MAIL.

Overall, this paper positions latent imagination as a principled and scalable substrate for imitation learning in complex environments where reward specification is difficult or impossible, and we hope it encourages further research at the intersection of world models and imitation learning.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. 1

[2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. 1

[3] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, 1995. 1, 2

[4] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 390–399. JMLR.org, 2017. 2, 5

[5] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 449–458. JMLR.org, 2017. 5

[6] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. 3

[7] Ivan Bratko, Tanja Urbančič, and Claude Sammut. Behavioural cloning: Phenomena, results and problems. *IFAC Proceedings Volumes*, 28(21):143–149, 1995. 5th IFAC Symposium on Automated Systems Based on Human Skill (Joint Design of Technology and Organisation), Berlin, Germany, 26-28 September. 1

[8] Silvia Chiappa, Sébastien Racanière, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *CoRR*, abs/1704.02254, 2017. 2

[9] Petros Christodoulou. Soft actor-critic for discrete action settings, 2019. 8

[10] Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. *arXiv preprint arXiv:2006.04678*, 2020. 2

[11] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning, 2018. 2, 4, 5

[12] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods, 2019. 4

[13] Adam Gleave, Mohammad Taufeeque, Juan Rocamonde, Erik Jenner, Steven H. Wang, Sam Toyer, Maximilian Ernestus, Nora Belrose, Scott Emmons, and Stuart Russell. imitation: Clean imitation learning implementations. arXiv:2211.11972v1 [cs.LG], 2022. 5

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2

[15] Karol Gregor, Danilo Jimenez Rezende, Frederic Besse, Yan Wu, Hamza Merzic, and Aaron van den Oord. Shaping belief states with generative environment models for rl. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[16] David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. 1, 2

[17] Danijar Hafner. Benchmarking the spectrum of agent capabilities, 2022. 5

[18] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *CoRR*, abs/1811.04551, 2018. 4

[19] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. 2

[20] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *CoRR*, abs/2010.02193, 2020. 2

[21] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. 2, 3, 4, 5

[22] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 1, 2, 4, 5

[23] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. 5

[24] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374, 2019. 2

[25] Ilya Kostrikov, Kumar Krishna Agrawal, Sergey Levine, and Jonathan Tompson. Addressing sample inefficiency and reward bias in inverse reinforcement learning. *CoRR*, abs/1809.02925, 2018. 1, 3, 4

[26] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. 2

[27] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 2

[28] Federico Malato, Florian Leopold, Andrew Melnik, and Ville Hautamäki. Zero-shot imitation policy via search in demonstration dataset. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024. 2

[29] Stephanie Milani, Anssi Kanervisto, Karolis Ramanauskas, Sander Schulhoff, Brandon Houghton, Sharada Mohanty, Byron Galbraith, Ke Chen, Yan Song, Tianze Zhou, Bingquan Yu, He Liu, Kai Guan, Yujing Hu, Tangjie Lv, Federico Malato, Florian Leopold, Amogh Raut, Ville Hautamäki, Andrew Melnik, Shu Ishida, João F. Henriques, Robert Klassert, Walter Laurito, Ellen Novoseller, Vinicius G. Goecks, Nicholas Waytowich, David Watkins, Josh Miller, and Rohin Shah. Towards solving fuzzy tasks with human feedback: A retrospective of the minerl basalt 2022 competition, 2023. 2, 6

[30] Yao Mu, Yuzheng Zhuang, Bin Wang, Guangxiang Zhu, Wulong Liu, Jianyu Chen, Ping Luo, Shengbo Eben Li, Chongjie Zhang, and Jianye Hao. Model-based reinforcement learning via imagination with derived memory. In *Advances in Neural Information Processing Systems*, 2021. 2

[31] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 1

[32] Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 4209–4215. IEEE Press, 2021. 2

[33] Manu Orsini, Anton Raichuk, Leonard Hussenot, Damien Vincent, Robert Dadashi, Sertan Girgin, Matthieu Geist, Olivier Bachem, Olivier Pietquin, and Marcin Andrychowicz. What matters for adversarial imitation learning? In *Advances in Neural Information Processing Systems*, pages 14656–14668. Curran Associates, Inc., 2021. 4, 3

[34] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models. *Neural Information Processing Systems*, 2021. 2

[35] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: imitation learning via regularized behavioral cloning. *CoRR*, abs/1905.11108, 2019. 5

[36] Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 661–668, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR. 1

[37] Claude Sammut, Scott Hurst, Dana Kedzier, and Donald Michie. Learning to Fly. In *International Conference on Machine Learning*, pages 385–393, 1992. 1, 2

[38] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999. 1

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 2, 5

[40] Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163, 1991. 1

[41] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. 1, 2, 3

[42] Ville Tanskanen, Arto Klami, and Ville Hautamäki. On the importance of representation in imitating human-like gameplay. In *CoG*, 2025. 2

[43] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in Neural Information Processing Systems*, 2015. 2

[44] Xingrui Yu, Yueming Lyu, and Ivor Tsang. Intrinsic reward driven imitation learning via generative model. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10925–10935. PMLR, 2020. 5

[45] Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in GANs. In *International Conference on Learning Representations*, 2018. 2

[46] Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl S. Muller, Jake A. Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset, 2019. 5

[47] Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl Muller, Jake Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6811–6820, 2020. 5

[48] Haibin Zhou, Tong Wei, Zichuan Lin, junyou li, Junliang Xing, Yuanchun Shi, Li Shen, Chao Yu, and Deheng Ye. Revisiting discrete soft actor-critic. *Transactions on Machine Learning Research*, 2024. 8