ACTIVE WORLD MODEL LEARNING WITH PROGRESS CURIOSITY

Kuno Kim[1], **Megumi Sano**[1], **Julian De Freitas**[5], **Nick Haber**[4,*] **& Daniel L. K. Yamins**[1,2,3,*]
Departments of Computer Science[1], Psychology[2], Wu Tsai Neurosciences Institute[3], and
Graduate School of Education[4], Stanford University
Department of Psychology[5], Harvard University
{kunokim}@stanford.edu

ABSTRACT

Infants appear to actively build models of their environment by attending to stimuli in a highly non-random manner. In this work, we study how to design such a curiosity-driven Active World Model Learning (AWML) system. To do so, we construct a curious agent building world models while visually exploring a 3D physical environment rich with distillations of representative real-world agents. We propose an AWML system driven by γ-Progress: a scalable and effective learning progress-based curiosity signal. We show that γ-Progress naturally gives rise to an exploration policy that directs attention to complex but learnable dynamics in a balanced manner, as a result overcoming the "white noise problem". As a result, our γ-Progress-driven controller achieves significantly higher AWML performance than state-of-the-art baselines.

1 INTRODUCTION

Infants appear to actively build models of their environment by attending to stimuli in a highly non-random manner (Smith et al., 2019; Gergely et al., 1995; Frankenhuis et al., 2013). With the aim of building a neural agent that can do the same, we study Active World Model Learning (AWML) – the problem of determining a directed exploration policy that enables efficient construction of better world models. (see Appendix B for formal definition) To do so, we construct a progress-driven curious neural agent performing AWML in a custom-built 3D virtual world environment. Specifically, our contributions are as follows: (1). We construct a 3D virtual environment rich with agents displaying a wide spectrum of realistic stimuli behavior types with varying levels of learnability, such as static, periodic, noise, peekaboo, chasing, and mimicry. (2). We propose an AWML system driven by γ-Progress: a novel and scalable learning progress-based curiosity signal. We show that γ-Progress gives rise to an exploration policy that overcomes the white noise problem (Schmidhuber, 2010) and achieves significantly higher AWML performance than state-of-the-art exploration strategies — including Random Network Distillation (RND) (Burda et al., 2018) and Model Disagreement (Pathak et al., 2019).

Related Works: A natural class of world models involve forward dynamics prediction. Action-conditioned forward models can be used directly in planning for robotic control tasks (Finn & Levine, 2017), as performance-enhancers for reinforcement learning tasks (Ke et al., 2019), or as "dream" environment simulations for training policies (Ha & Schmidhuber, 2018). A key question the agent is faced with is how to choose its actions to efficiently learn the world model. One approach is to pursue *novelty*, e.g

[*] equal contribution

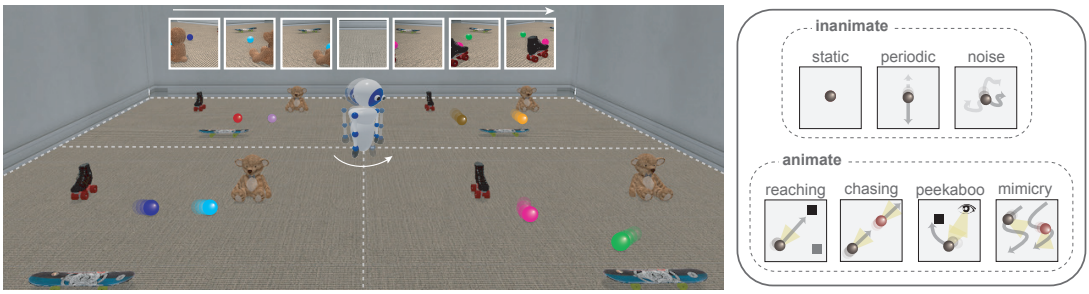


Figure 1: **Virtual environment.** Our 3D virtual environment is a distillation of key aspects of real-world environments. The *curious agent* (white robot) is centered in a room, surrounded by various *external agents* (colored spheres) contained in different quadrants, each with dynamics that correspond to a realistic inanimate or animate behavior (right box). The curious agent can rotate to attend to different behaviors as shown by the first-person view images at the top. See https://bit.ly/31vg7v1 for videos.

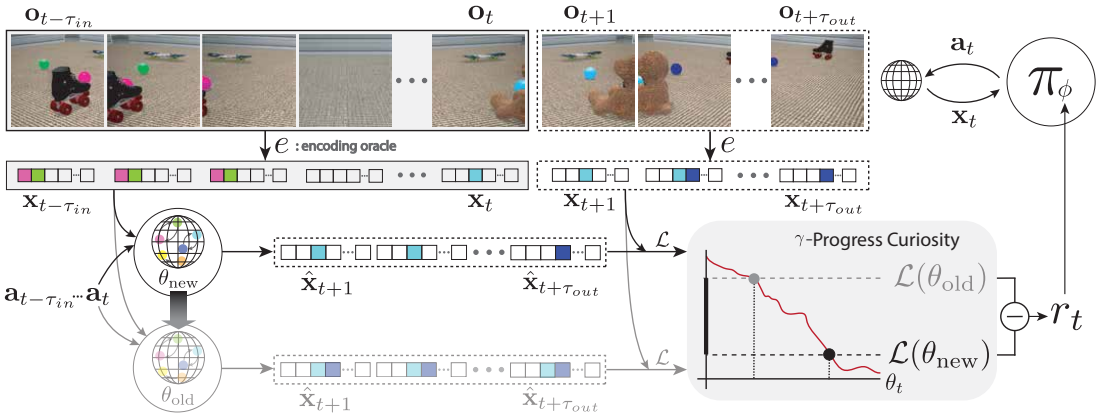


Figure 2: **Active World Model Learning with γ-Progress** The curious agent consists of a *world model* and a *progress-driven controller*. Both the new (black) and old (gray) models take as input object-oriented features $\mathbf{x}_{t-\tau_{in}:t}$ and predict $\hat{\mathbf{x}}_{t:t+\tau_{out}}$. The old model weights, θ_{old}, are slowly updated to the new model weights θ_{new}. The controller, π_ϕ, is optimized to maximize γ-Progress reward, i.e $\mathcal{L}(\theta_{old}) - \mathcal{L}(\theta_{new})$.

count-based and pseudo-count-based methods (Strehl & Littman, 2008; Bellemare et al., 2016; Ostrovski et al., 2017), Random Network Distillation (RND) (Burda et al., 2018), and *empowerment* (Mohamed & Rezende, 2015). Another fundamental approach is to use *adversarial* curiosity, which take actions estimated to maximize world model prediction error (Stadie et al., 2015; Pathak et al., 2017; Haber et al., 2018). However, adversarial curiosity is especially prone to the *white noise problem*, in which agents are motivated to waste time fruitlessly trying to solve unsolvable world model problems. (Schmidhuber, 2010) Estimating learning progress (Oudeyer et al., 2007; 2013; Achiam & Sastry, 2017) or *information gain* (Houthooft et al., 2016) avoids the white noise problem in a more comprehensive fashion. However, such methods have been limited in scope because they involve high computational costs. In this work, we present a novel method for estimating learning progress in a computationally scalable fashion.

2 Virtual World Environment

We design our 3D virtual environment to preserve the following key properties of real-world environments: *diverse dynamics*, i.e containing various agent-specific programs, *partial observability*, information limited to what lies within view, and *interactivity*, agent's actions influence the world. Our virtual environment consists of two main components, a *curious neural agent* and various *external agents*.

The **curious neural agent**, embodied by an avatar, fixed at the center of a room (Figure 1). Just as a human toddler can control her gaze to visually explore her surroundings, the agent is able to partially observe the environment based on what lies in its field of view (see top of Figure 1). The agent can choose from 9 actions: rotate $12°, 24°, 48°$, or $96°$, to the left/right, or stay in its current orientation. The **external agents** are spherical avatars that each act under a policy inspired by real-world inanimate and animate stimuli. An *external agent behavior* consists of either one external agent, e.g reaching, or two interacting ones, e.g chasing. Since external agents are devoid of surface features, the curious agent must learn to attend to different behaviors based on spatiotemporal kinematics alone. We experiment with external agent behaviors (see Figure 1, right) including static, periodic, noise, reaching, chasing, peekaboo, and mimicry. (See Appendix A for details) The animate behaviors have deterministic and stochastic variants, where the stochastic variant preserves the core dynamics underlying the behavior, albeit with more randomness. See https://bit.ly/31vg7v1 for video descriptions of the environment and external agent behaviors.

We divide the room into four quadrants, each of which contains various auxiliary objects (e.g teddy bear, roller skates, surfboard) and one external agent behavior. The room is designed such that the curious agent can see at most one external agent behavior at any given time.

3 Methods

We describe a practical instantiation of the two components of our curious neural agent: a *world model* fitting the forward dynamics and a *progress-driven controller* which acts to maximize γ-Progress reward.

World Model. We assume that the agent has access to an oracle encoder $e : \mathcal{O} \to \mathcal{X}$ that maps an image observation $\mathbf{o}_t \in \mathcal{O}$ to a disentangled object-oriented feature vector $\mathbf{x}_t = (\mathbf{x}_t^{ext}, \mathbf{x}_t^{aux}, \mathbf{x}_t^{ego})$ where

$\mathbf{x}_t^{ext} = (\tilde{\mathbf{c}}_t, \mathbf{m}_t) = (\tilde{\mathbf{c}}_{t,1}, \ldots \tilde{\mathbf{c}}_{t,n_{ext}}, \mathbf{m}_{t,1}, \ldots, \mathbf{m}_{t,n_{ext}})$ contains information about the external agents; namely the observability masks $\mathbf{m}_{t,i}$ ($\mathbf{m}_{t,i} = 1$ if external agent i is in curious agent's view at time t, else $\mathbf{m}_{t,i} = 0$) and masked position coordinates $\tilde{\mathbf{c}}_{t,i} = \mathbf{c}_{t,i}$ if $\mathbf{m}_{t,i} = 1$ and else $\tilde{\mathbf{c}}_{t,i} = \hat{\mathbf{c}}_{t,i}$. Here, $\mathbf{c}_{t,i}$ is the true global coordinate of external agent i and $\hat{\mathbf{c}}_{t,i}$ is the model's predicted coordinate of external agent i where $i = 1, \ldots, n_{ext}$. $\mathbf{x}_t^{aux}$ contains coordinates of auxiliary objects, and $\mathbf{x}_t^{ego}$ contains the ego-centric orientation of the curious agent. Our world model ω_θ is an ensemble of component networks $\{\omega_{\theta^k}\}_{k=1}^{N_{\text{cc}}}$ where each ω_{θ^k} independently predicts the forward dynamics for a subset $I_k \subseteq \{1, ..., \dim(\mathbf{x}^{ext})\}$ of the input dimensions of $\mathbf{x}^{ext}$ corresponding to a minimal interdependent group in the world. For example, $\mathbf{x}^{ext}_{t:t+\tau,I_k}$ may correspond to the masked coordinates and observability masks of the chaser and runner external agents for times $t, t+1, ..., t+\tau$. We assume $\{I_k\}_{k=1}^{n_{cc}}$ is given as prior knowledge but future work may integrate disentanglement learning into our pipeline. A component network ω_{θ^k} takes as input $(\mathbf{x}^{ext}_{t-\tau_{in}:t,I_k}, \mathbf{x}^{aux}_{t-\tau_{in}:t}, \mathbf{x}^{ego}_{t-\tau_{in}:t}, \mathbf{a}_{t-\tau_{in}:t+\tau_{out}})$, where $\mathbf{a}$ denotes the curious agent's actions, and outputs $\hat{\mathbf{x}}^{ext}_{t:t+\tau_{out},I_k}$. The outputs of the component network are concatenated to get the final output $\hat{\mathbf{x}}^{ext}_{t:t+\tau_{out}} = (\hat{\mathbf{c}}_{t:t+\tau_{out}}, \hat{\mathbf{m}}_{t:t+\tau_{out}})$. The world model loss is:

$$\mathcal{L}(\theta, \mathbf{x}_{t-\tau_{in}:t+\tau_{out}}, \mathbf{a}_{t-\tau_{in}:t+\tau_{out}}) = \sum_{t'=t}^{t+\tau_{out}} \sum_{i=1}^{N_{ext}} \mathbf{m}_{t',i} \cdot \|\hat{\mathbf{c}}_{t',i} - \tilde{\mathbf{c}}_{t',i}\|_2 + \mathcal{L}_{ce}(\hat{\mathbf{m}}_{t',i}, \mathbf{m}_{t',i}) \quad (1)$$

where $\mathcal{L}_{ce}$ is cross-entropy loss. We parameterize each component network ω_{θ^k} with a two-layer Long Short-Term Memory (LSTM) network followed by two-layer Multi Layer Perceptron (MLP). The number of hidden units are adapted to the number of external agents being modeled.

Progress-driven Controller. We propose γ**-Progress**, a scalable progress-based curiosity signal which approximates learning progress by the difference in the losses of an old model and a new model. The old model weights, θ_{old}, lag behind those of the new model, θ_{new}, with a simple update rule: $\theta_{old} \leftarrow \gamma\theta_{old} + (1-\gamma)\theta_{new}$, where γ is scalar mixing constant. The curiosity reward is:

$$R(\mathbf{x}_t) = \mathcal{L}(\theta_{new}, \mathbf{x}_{t-\tau_{in}-\tau_{out}:t}, \mathbf{a}_{t-\tau_{in}-\tau_{out}:t}) - \mathcal{L}(\theta_{old}, \mathbf{x}_{t-\tau_{in}-\tau_{out}:t}, \mathbf{a}_{t-\tau_{in}-\tau_{out}:t}) \quad (2)$$

Our controller π_ϕ follows an ϵ-*greedy* sampling scheme with respect to a Q-function Q_ϕ trained with the curiosity reward in Eq. 2. Q_ϕ is parametrized by a two-layer MLP with 512 hidden units that takes as input $\mathbf{x}_{t-2:t}$ and outputs estimated state-action values for all nine possible actions. Q_ϕ is updated with the DQN Mnih et al. (2013) learning algorithm.

4 Results

We evaluate the AWML performance of γ-Progress on two metrics: *end performance* and *sample complexity*. End performance is the inverse of the the final world loss after a larger number of environment interactions, and intuitively measures the "consistency" of the proxy reward with respect to the true reward. Sample complexity measures the rate of reduction in world model loss $\mathcal{L}_\mu(\theta)$ with respect to the number of environment interactions. The samples from the validation distribution μ correspond to core validation cases we crafted for each behavior. For details, see Appendix F. Experiments are run in the Mixture world where the virtual environment is instantiated external agents spanning four representative types: static, periodic, noise, and animate. This set up is a natural distillation of a real-world environment containing a

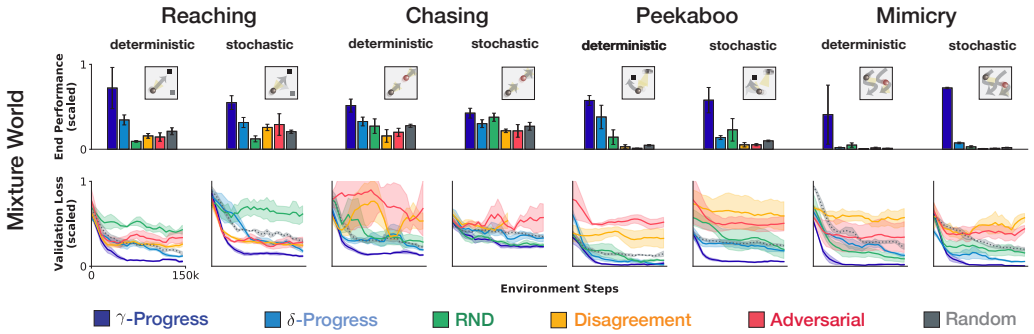


Figure 3: **AWML Performance**. The animate external agent is varied across experiments according to the column labels. Error bars/regions are standard errors of the best 5 seeds out of 10. γ-Progress achieves lower sample complexity than all baselines on $7/8$ behaviors. Notably, γ-Progress also outperforms all baselines in end performance on $6/8$ behaviors.

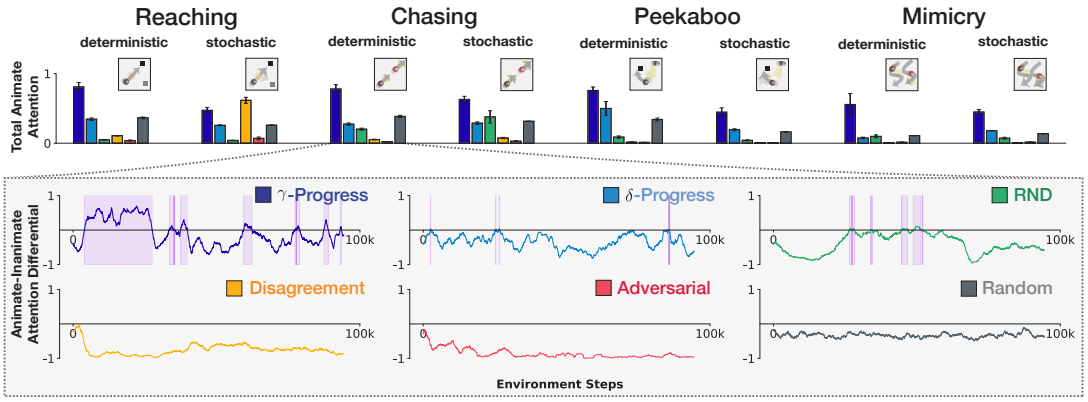


Figure 4: **Attention Patterns**. Bar plots show total animate attention, i.e ratio between the number of time steps an animate external agent was visible to the curious agent, and the time steps a noise external agent was visible. Time series plots (zoom-in box) show the differences between mean attention to the animate external agents and the mean of attention to other agents in a 500 step window, with periods of animate preference highlighted in purple. Results averaged across 5 runs. γ-Progress displays strong animate attention while baselines are indifferent, e.g δ-Progress, or fixating on white noise, e.g Adversarial.

wide spectrum of behaviors. We run separate experiments in which the animate external agents are varied amongst the deterministic and stochastic versions of reaching, chasing, peekaboo, and mimicry agents (see Section 2). We compare the AWML performance of the following methods:

γ-Progress (Ours) is our proposed variant of progress curiosity which chooses θ_{old} to be a geometric mixture of all past models as in Eq. 11.

δ-Progress (Achiam & Sastry, 2017; Graves et al., 2017) is the δ-step learning progress reward from Eq. 10 with $\delta = 1$. We found that any $\delta > 3$ is impractical due to memory constraints.

RND (Burda et al., 2018) is a novelty-based method that trains a predictor neural net to match the outputs of a random state encoder. States for which the predictor networks fails to match the random encoder are deemed "novel", and thus receive high reward.

Disagreement (Pathak et al., 2019) is the disagreement based method from Eq. 6 with $N = 3$ ensemble models. We found that $N > 3$ is impractical due to memory constraints.

Adversarial (Stadie et al., 2015; Pathak et al., 2017) is the prediction error based method from Eq. 5. We use the ℓ_2 prediction loss of the world model as the reward.

Random chooses actions uniformly at random among the 9 possible rotations.

Fig. 3a shows end performance (first row) and sample complexity (second row) in the Mixture world. In the Mixture world, we see that γ-Progress has lower sample complexity than δ-Progress, Disagreement, Adversarial, and Random baselines on all $8/8$ behaviors and outperforms RND on $7/8$ behaviors while tying on stochastic chasing. See `https://bit.ly/31vg7v1` for visualizations of model predictions.

Figure 4 shows the ratio of attention to animate vs other external agents for each behavior in the Mixture world as well as example animate-inanimate attention differential timeseries. The γ-Progress agents spend substantially more time attending to animate agents than do alternative policies. This increased animate-inanimate attention differential often corresponds to a characteristic attentional "bump" that occurs early as the γ-Progress curious agent focuses on animate external agents quickly before eventually "losing interest" as prediction accuracy is achieved. Strong animate attention emerges for $7/7$ behaviors when using γ-Progress. Baselines display two distinct modes that lead to lower performance (Figure 4, bottom). The first is *attentional indifference*, in which it finds no particular external agent interesting. δ-Progress frequently had attentional indifference as the new and old world model, separated by a fixed time difference, were often too similar to generate a useful curiosity signal. The second failure mode is *white noise fixation*, where the observer is captivated by the noise external agents. RND suffers from white noise fixation due to the fact that our noise behaviors have the most diffuse visited state distribution. We also observe that for noise behaviors, a world model ensemble does not collectively converge to a single mean prediction, and as a result Disagreement finds the noise behavior highly interesting. Finally, the Adversarial baseline fails since noise behaviors yield the highest prediction errors. The white noise failure mode is particularly detrimental to sample complexity, with RND, Disagreement, and Adversarial, as evidenced by their below-Random performance.

REFERENCES

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *arXiv preprint arXiv:1705.10528*, May 2017.

Janet W Astington, Paul L Harris, and David R Olson. *Developing theories of mind*. CUP Archive, 1990.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv preprint arXiv:1605.08803*, May 2016.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2786–2793. IEEE, 2017.

Willem E Frankenhuis, Bailey House, H Clark Barrett, and Scott P Johnson. Infants' perception of chasing. *Cognition*, 126(2):224–233, 2013.

György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56(2):165–193, 1995.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1311–1320. JMLR. org, 2017.

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

Nick Haber, Damian Mrowca, Stephanie Wang, Li Fei-Fei, and Daniel LK Yamins. Learning to play with intrinsically-motivated self-aware agents. In *Advances in Neural Information Processing Systems*, 2018.

Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1109–1117. Curran Associates, Inc., 2016.

Nan Rosemary Ke, Amanpreet Singh, Ahmed Touati, Anirudh Goyal, Yoshua Bengio, Devi Parikh, and Dhruv Batra. Learning dynamics model in reinforcement learning by incorporating the long term future. *arXiv preprint arXiv:1903.01599*, 2019.

Cam Linke, Nadia M Ady, Martha White, Thomas Degris, and Adam White. Adapting behaviour via intrinsic reward: A survey and empirical study. *arXiv preprint arXiv:1906.07865*, 2019.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2721–2730. JMLR. org, 2017.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Pierre-Yves Oudeyer, Adrien Baranes, and Frédéric Kaplan. Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints. In *Intrinsically motivated learning in natural and artificial systems*, pp. 303–365. Springer, 2013.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *arXiv preprint arXiv:1705.05363*, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. *arXiv:1906.04161*, 2019.

David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990 – 2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, Sept 2010.

Burr Settles. *Active Learning*, volume 18. Morgan & Claypool Publishers, 2011.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294. ACM, 1992.

Kevin Smith, Lingjie Mei, Shunyu Yao, Jiajun Wu, Elizabeth Spelke, Josh Tenenbaum, and Tomer Ullman. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. In *Advances in Neural Information Processing Systems*, pp. 8983–8993, 2019.

Bradly Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

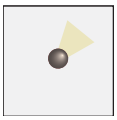
Henry M Wellman. *The child's theory of mind.* The MIT Press, 1992.

Appendix - Active World Model Learning in Agent-rich Environments with Progress Curiosity

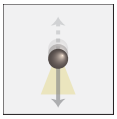
A External Agent Behaviors

Below, we describe all behaviors in detail. Note that the animate behaviors (peekaboo, reaching, chasing, and mimicry) are further sub-divided into deterministic and stochastic versions.

Inanimate behaviors



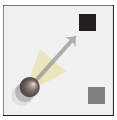
Static Inspired by stationary objects such as couches, lampposts, and fire hydrants, the *static agent* remains at its starting location and stays immobile.



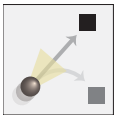
Periodic Inspired by objects exhibiting periodic motion such as fans, flashing lights, and clocks, the *periodic agent* regularly moves back and forth between two specified locations in its quadrant.

Noise Inspired by random motion in wind, water, and other inanimate elements, the *noise agent* randomly samples a new direction and moves in that direction with a fixed step size while remaining within the boundaries of its quadrant.

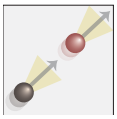
Animate Behaviors



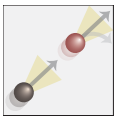
Reaching (deterministic) We often exhibit goal-oriented behavior by interacting with objects. The *reacher agent* approaches each auxiliary object in its quadrant sequentially, such that object positions fully determine its trajectory. Objects periodically shift locations such that predicting agent behavior at any given time requires knowing the current object positions.



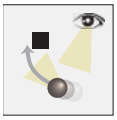
Reaching (stochastic) The order in which the reacher agent visits the objects is stochastic (uniform sampling from the three possible objects). However, once the reacher agent starts moving towards an object, its trajectory for the next few time steps, before it chooses a different object to move to, is predictable.



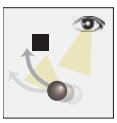
Chasing (deterministic) We often act contingently on the actions of other agents, which in turn depend on our own. In chasing, a *chaser agent* chases a *runner agent*. If the runner is too close to quadrant bounds, it then escapes to one of a few escape locations away from the chaser but within the quadrant. Thus, the chaser's position affects the runner's trajectory and vice versa.



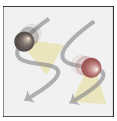
Chasing (stochastic) When the runner agent is too close to the quadrant bounds, it escapes by picking any random location away from the chaser and within the bounds of the quadrant.



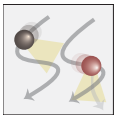
Peekaboo (deterministic) One way of detecting an animate agent is if its motion is contingent on our own. The *peekaboo agent* acts contingently on the curious agent. If the curious agent stares at it, it hides behind an auxiliary object such as a doll. If the curious agent continues to stare, it starts *peeking* out by moving to a fixed peek location. If the curious agent looks away, it stops hiding, returning to its exposed location.



Peekaboo (stochastic) There are multiple peeking locations near the hiding object that the peekaboo agent can visit randomly during its peeking behavior.



Mimicry (deterministic) From an early age, we learn by imitating others. Mimicry consists of an *actor agent* and an *imitator agent*, each staying in one half of the quadrant to avoid collisions. The actor acts identically to the random agent, while the imitator mirrors the actor's trajectory with a delay, such that the past trajectory of the actor fully determines the future trajectory of the imitator.



Mimicry (stochastic) The imitator agent is imperfect and produces a noisy reproduction of the actor agent's trajectory.

B THEORY

In this section we formalize Active World Model Learning (AWML) as a Reinforcement Learning (RL) problem that is a specific form of active learning. We then discuss a number of curiosity signals that can be used to drive AWML, and introduce γ-Progress, a scalable progress-based measure with several algorithmic and computational advantages over previous signals.

B.1 ACTIVE WORLD MODEL LEARNING

We formalize an agent in environment as the tuple $\mathcal{E} := (\mathcal{S}, \mathcal{A}, P, P_0)$. $\mathcal{S}$ denotes the set of states the agent and environment can be in — in the virtual world environment described in section 2, $\mathcal{S}$ captures the gaze direction of the curious agent, the positions and type of external objects, and the positions and internal states of the external agents[1]. $\mathcal{A}$ represents the set of actions the agent can take, and are constrained by the physical avatar of the agent — in the virtual world, the choice of how far and where to turn its gaze. Transition dynamics are given by the function $P : \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$, where $\Omega(\mathcal{S})$ is the set of probability measures on $\mathcal{S}$ (allowing for stochastic environment dynamics). In the case of our virtual world, P captures both the effect of the gaze actions of the agent (e.g. changes in which part of the scene is being observed), as well the dynamics of each of the external agents. The function $P_0 : \mathcal{S} \to [0, 1]$ describes the probability distribution of initial conditions of states.

In this environment, the agent's overall goal is to learn a target function ω with as few data samples as possible. In general, ω can be any predictor on finite-horizon state-action trajectories sampled from the environment. That is, $\omega : \mathcal{X} \to \Omega(\mathcal{Y})$, where $\mathcal{X} := \mathcal{S}^{i_s} \times \mathcal{A}^{i_a}$ and $\mathcal{Y} := \Omega(\mathcal{S}^{o_s} \times \mathcal{A}^{o_a})$ represent sets of fixed-length observation-action sequences. (The non-negative integers i_s, i_a, o_s, and o_a are the input and output state and action horizons, respectively.) In this work, we work with forward prediction, i.e. the situation where $\mathcal{X} = \mathcal{S} \times \mathcal{A}, \mathcal{Y} = \mathcal{S}$, and $\omega = P$[2], but a variety of other potentially useful targets, such as inverse prediction, can also be formulated by appropriate choice of $\mathcal{X}, \mathcal{Y}$ and ω.

The agent seeks to estimate a parameterized model ω_θ of ω (e.g θ are parameters deep neural network; see section 3 below). We henceforth refer to ω_θ as the world model. To measure its error during world model optimization, the agent is equipped with a loss function $\mathcal{L} : (x, f, g) \mapsto \mathbb{R}$ such that for any $x \in \mathcal{X}$ and any functions $f, g : \mathcal{X} \to \Omega(\mathcal{Y})$, $\mathcal{L}(x, f, g)$ achieves its minimum whenever $f(x) = g(x)$. A measure μ over $\mathcal{X}$ representing a validation data distribution is also specified, so that the agent's learning goal is to minimize $\mathcal{L}_\mu(\theta) := \mathbb{E}_\mu[\mathcal{L}(\theta)] = \int_{\mathcal{X}} \mathcal{L}(x, \omega(x), \omega_\theta(x))\mu(x)dx$.

The agent learns the world model from data gathering by acting in the environment. We formally define Active World Model Learning as a Markov Decision Process (MDP) $\mathcal{M} := (\bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{P}, \bar{P}_0, r)$ with state and action spaces $\bar{\mathcal{S}}, \bar{\mathcal{A}}$, dynamics and initial conditions $\bar{P}, \bar{P}_0$, and reinforcement reward function r. Because intrinsically-motivated policies (such as progress curiosity) will critically depend on states of the agent's world model, $\mathcal{M}$ is an augmentation of the environment $\mathcal{E}$ that is constructed by adding the data-collection and model parameter history of the agent itself.

Specifically, the augmented state space $\bar{\mathcal{S}} := \mathcal{S} \times \mathcal{H} \times \Theta$, so that $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$ has the form $\bar{\mathbf{s}} = (\mathbf{s}, H, \theta)$. $\mathbf{s} \in \mathcal{S}$ is an environment state, $H = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1 \ldots) \in \mathcal{H}$ is the history of environment state-actions visited so far, and $\theta \in \Theta$ is the current model parameters. The action space $\bar{\mathcal{A}} := \mathcal{A}$ is simply the same set of actions available to the agent in the environment[3]. The dynamics are described by $\bar{P} : \bar{\mathcal{S}} \times \mathcal{A} \to \Omega(\bar{\mathcal{S}})$, which step $\mathbf{s}$ according to the environment dynamics P, augment the history with new data, and updates

[1] Our virtual world environment is *partially observable* and hence requires the additional specification of $\mathcal{O}$, the set of observations, and $Q = Q(\mathbf{o}|\mathbf{s}, \mathbf{a})$, the set of conditional observation probabilities. For the sake of simplicity, we suppress this complication in the main text and point out where it is salient in a series of footnotes.

[2] In a partial observable case such as ours, the agent predicts observations from a sequence of past observations, which contains additional state information (e.g. the direction an external agent is moving) relevant to predicting the next observation. This state information can be incomplete (e.g. if an external agent is invisible until the observation to be predicted), leading to what might be thought of as additional white noise, or *degeneracy* in the world model problem (Haber et al., 2018).

[3] In the partial observability case, the action choice determines not only the state transition but also what is observable each timestep, and hence the agent should keep the interesting in view. The MDP becomes a POMDP, where we assume that the agent has full access to its internal state and history, so augmented observations $\bar{o} \in \bar{\mathcal{O}} = \mathcal{O} \times \mathcal{H} \times \Theta$

the world model ω_θ on the augmented history. Formally this is described by the sampling procedure:

$$(\mathbf{s}', H', \theta') \sim \bar{P}(\cdot|\bar{\mathbf{s}} = (\mathbf{s}, H, \theta), \mathbf{a}) \text{ where } \mathbf{s}' \sim P(\mathbf{s}, \mathbf{a}),\ H' = H \cup \{\mathbf{a}, \mathbf{s}'\},\ \theta' \sim P_\ell(H', \theta)$$

where $P_\ell : H \times \Theta \to \Omega(\Theta)$ is (stochastic) update rule for the world model parameters, e.g. a (stochastic) learning algorithm which updates the parameters on the history of data. The initial conditions $\bar{P}_0(\bar{\mathbf{s}} = (\mathbf{s}, H, \theta)) = P_0(\mathbf{s})\mathbb{1}(H = \{\})q(\theta)$ is the augmented initial-distribution where $\mathbb{1}$ is the indicator function and $q(\theta)$ is a prior distribution over the model parameters.

The function r encodes the learning objective of the agent as an RL reward. A policy is a map $\pi : \mathcal{S} \to \Omega(\mathcal{A})$ from states to action distributions. In general, the infinite-horizon RL problem is to find an optimal policy $\pi^* = \arg\max_\pi J(\pi)$, where $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \beta^t r_t]$ and $0 \leq \beta < 1$ is a discount factor. The goal of AWML in specific is to make effective data-collection decisions to minimize world model loss. This could in theory by accomplished by taking the reward function of AWML to be

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = -\mathcal{L}_\mu(\theta'),$$

where $\bar{\mathbf{s}} = (\mathbf{s}, H, \theta), \bar{\mathbf{s}}' = (\mathbf{s}', H', \theta')$ and $\theta' = P_\ell(H \cup \{\mathbf{a}, \mathbf{s}'\}, \theta)$ is the updated model parameters after collecting new data $\{\mathbf{a}, \mathbf{s}'\}$. Given the definition of total reward J, this is equivalent, up to monotonic transform, to the reward function:

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}_\mu(\theta) - \mathcal{L}_\mu(\theta'). \tag{3}$$

Defined thusly, $r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}')$ measures the reduction in world model loss as a result of obtaining new data $\{\mathbf{a}, \mathbf{s}'\}$, i.e the *prediction gain*.

By appropriately constructing $\mathcal{M}$, different variants of traditional active learning can be recovered as AWML problems. For example, Query Synthesis Active Learning (Settles, 2011) is obtained by taking $\mathcal{S} = \mathcal{Y}, \mathcal{A} = \mathcal{X}$, and $P(\cdot|\cdot, \mathbf{a} = \mathbf{x}) = \omega(\mathbf{x})$. In words, the agent proposes a synthetic data query $\mathbf{a}$ and the oracle P provides a label $\mathbf{s}'$. Other traditional active learning tasks can also be derived, including pool-based and stream active learning (see Appendix C for details).

However, there are several complications making it challenging to use equation 3 directly. First, μ can be a rather diffuse distribution which makes it intractable to compute equation 3 at every environment step. This is especially problematic in the types of environments of interest here and in other recent works on curiosity-driven learning, relative to the more constrained situations of traditional active learning. Secondly, in cases in which an agent explores an unknown environment, μ is not even known prior to interacting with the environment. These bottlenecks necessitate an efficiently-computable heuristic reward function that will typically promote the same learning goal of equation 3 — constructing a learning dataset that minimizes the loss $\mathcal{L}_\mu$ — while being independent of any particular choice of μ. The literature on algorithmic curiosity has explored many variants of such heuristic "curiosity signals", which achieve consistency with the learning goals of equation 3 with varying degrees of accuracy and efficiency. A spectrum of such ideas, including our novel proposal (γ-Progress), are described in the next section.

B.2 Curiosity Signals

We now motivate γ-Progress by outlining the limitations of previously proposed curiosity signals and highlighting the computational and algorithmic advantages of our method.

Information Gain (Houthooft et al., 2016; Linke et al., 2019) based methods seek to minimize uncertainty in the Bayesian posterior distribution over model parameters:

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = D_{\text{KL}}(p(\theta')||p(\theta)) \tag{4}$$

where $p(\theta') = p(\theta|H \cup \{\mathbf{a}, \bar{\mathbf{s}}'\})$ and $p(\theta) = p(\theta|H)$. Note that, information gain is a lower bound to the prediction gain under weak assumptions (Bellemare et al., 2016). If the posterior has a simple form such as Laplace or Gaussian, information gain can be estimated by weight change $|\theta' - \theta|$ (Linke et al., 2019), and otherwise one may resort to learning a variational approximation q to approximate the information gain with $D_{\text{KL}}(q(\theta')||q(\theta))$ (Houthooft et al., 2016). The former weight change methods require a model after every step in the environment and is thus impractical in many settings where world model updates are expensive, e.g. backpropagation through deep neural nets. The latter family of variational methods require maintenance of a parameter distribution and an interlaced evidence lower bound optimization and are thus impractical to use with modern deep nets (Achiam & Sastry, 2017).

has the form $\bar{o} = (o, H, \theta)$, where $o \in O$ (augmented conditional observation probabilities $\bar{Q}$ are similarly derived from Q).

Adversarial (Stadie et al., 2015; Pathak et al., 2017; Haber et al., 2018) curiosity assumes prediction gain is proportional to the current world model loss, which, for forward prediction AWML with negative log likelihood loss, is

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = -\log \omega_\theta(\mathbf{s}'|\mathbf{s}, \mathbf{a}). \tag{5}$$

This assumption holds when the target function ω is learnable by the model class Θ and the learning algorithm P_ℓ makes monotonic improvement without the need for curriculum learning. However, adversarial reward is perpetually high when the target is unlearnable by the model class, e.g. deterministic model ω_θ cannot match stochastic target ω on inputs $\mathbf{x}$ for which $\omega(\mathbf{x})$ is not a Dirac-delta function. As a result, the curious agent suffers from the white noise problem (Schmidhuber, 2010), i.e it endlessly fixates on unlearnable stimuli.

Disagreement (Pathak et al., 2019) assumes future world model loss reduction is proportional to the prediction variance of an ensemble of N world models $\{P_{\theta_j}\}_{j=1}^N$.

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \text{Var}(\{\omega_{\theta_j}(\mathbf{s}'|\mathbf{s}, \mathbf{a})\}_{j=1}^N) \tag{6}$$

This approximation is reasonable when there exists a unique optimal world model. As we will show, for complex target functions all members of the ensemble do not converge to a single model and as a result the white noise problem persists. A key limitation of this method is that memory usage grow linearly with size of the model ensemble. Disagreement-based curiosity is known as query by committee sampling (Seung et al., 1992) in active learning.

Novelty (Bellemare et al., 2016; Dinh et al., 2016; Burda et al., 2018) methods reward transitions with a low visitation count $\mathcal{N}(s, a, s')$. The prototypical novelty reward is:

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{N}(s_t, a_t)^{-1/2} \tag{7}$$

Bellemare et al. (2016) generalize visitation counts to pseudocounts for use in continuous state, action spaces. Novelty is a good surrogate reward when one seeks to maximize coverage over the transition space regardless of the learnability of the transition. This characteristic makes novelty reward prefer noisy data drawn from a high entropy distribution. Novelty reward is not adapted to the world model and thus has a propensity to be inefficient at reducing world model loss.

Progress (Schmidhuber, 2010; Achiam & Sastry, 2017; Graves et al., 2017) The key idea is to simply approximate the expectation involving μ in equation 3 with the prediction gain on the history.

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}_{H'}(\theta) - \mathcal{L}_{H'}(\theta') \tag{8}$$

where H' is the augmented history after adding $(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}')$. There is no guarantee the optimal policy with respect to equation 8 is also an optimal policy with respect to equation 3 for every choice of μ. However, we expect this history-based approximation of prediction gain to generate a data distribution that will be suitable for a wide array of μ. If we think of the target ω as having easy, hard but doable, and impossible instances $(\mathbf{x}, \mathbf{y})$, we expect such an agent to spend some time sampling easy, a good deal of time sampling the hard but doable, and little time on the impossible. For μ with support on easy data, little sampling is needed; for support on hard but doable, the greater proportion of samples is useful; and support on the impossible does not contribute to Equation 3. Intuitively (if not formally), the progress curiosity approach should thus yield a data distribution that is proportionate to the intrinsic learnability of the target ω.

To ensure that the data generated and acted upon by equation 8 be a representative sample of the possible distribution of states — i.e. is as effective an approximation of prediction gain as possible — it is useful to pool data across multiple timepoints, including as the world model itself changes. This comes at the cost, however, of requiring access to model parameters at those multiple timepoints. Ensuring reliable and efficient approximation of progress requires careful choices of how often to update θ' and how to integrate information across multiple updates.

One approach to such choices is given by δ-**progress** (Achiam & Sastry, 2017; Graves et al., 2017), measures how much better the current "new" model θ_{new} is compared to an old model θ_{old}, which, for forward prediction AWML, is

$$r(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \log \frac{\omega_{\theta'}(\mathbf{s}'|\mathbf{s}, \mathbf{a})}{\omega_\theta(\mathbf{s}'|\mathbf{s}, \mathbf{a})} \simeq \log \frac{\omega_{\theta_{new}}(\mathbf{s}'|\mathbf{s}, \mathbf{a})}{\omega_{\theta_{old}}(\mathbf{s}'|\mathbf{s}, \mathbf{a})}. \tag{9}$$

Recall that μ is ideally a distribution whose support is learnable data with respect to model class Θ. There are two steps of approximation in equation 9. The first step assumes that training on a sample $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ affects the total validation loss on learnable data μ only through the reduction in loss on that particular sample. The second step assumes that future prediction gain is close to past prediction gain measured

with respect to $\theta_{new}, \theta_{old}$. The choice of $\theta_{new}, \theta_{old}$ is crucial to the efficacy of the progress reward. A popular approach (Achiam et al., 2017; Graves et al., 2017) is to choose

$$\theta_{new} = \theta_k, \quad \theta_{old} = \theta_{k-\delta}, \quad \delta > 0 \tag{10}$$

where θ_k is the model parameter after k update steps using P_ℓ. Intuitively, if the progress horizon δ is too large, we obtain an overly optimistic approximation of future progress. However if δ is too small, the agent may prematurely give up on learning hard transitions, e.g. where the next state distribution is very sharp. In practice, tuning the value δ presents a major challenge. Furthermore, the widely pointed out (Pathak et al., 2019) limitation of δ-Progress is that the memory usage grows $\mathcal{O}(\delta)$, i.e one must store δ world model parameters $\theta_{k-\delta}, ..., \theta$. As a result it is intractable in practice to use $\delta > 3$ with deep neural net models.

Here we propose γ**-Progress**, the following choice of $\theta_{new}, \theta_{old}$ to overcome both hurdles faces by δ-progress:

$$\theta_{new} = \theta, \quad \theta_{old} = (1-\gamma)\sum_{i=1}^{k-1} \gamma^{k-1-i}\theta_0 \tag{11}$$

In words, the old model is a weighted mixture of past models where the weights are exponentially decayed into the past. γ-Progress can be interpreted as a noise averaged progress signal. Conveniently, γ-Progress can be implemented with a simple θ_{old} update rule:

$$\theta_{old} \leftarrow \gamma\theta_{old} + (1-\gamma)\theta_{new} \tag{12}$$

Similar to equation 10, we may control the level of optimism towards expected future loss reduction by controlling the progress horizon γ, i.e a higher γ corresponds to a more optimistic approximation. γ-Progress has key practical advantages over δ-Progress: γ is far easier to tune than δ, e.g. we use a single value of γ throughout all experiments, and memory usage is constant with respect to γ. Crucially, the second advantage enables us to tune the progress horizon so that the model does not prematurely give up on exploring hard transitions. The significance of these practical advantages will become apparent from our experiments.

C Connections between General Active Learning and Conventional Active Learning

Query Synthesis Active Learning is obtained by taking $\mathcal{S} = \mathcal{Y}, \mathcal{A} = \mathcal{X}, P(\cdot|\cdot, \mathbf{a} = \mathbf{x}) = \omega(\mathbf{x})$ and $c(\bar{\mathbf{s}} = (\mathbf{s}, H, \theta), \mathbf{a}, \bar{\mathbf{s}}' = (\mathbf{s}', H', \theta')) = \mathcal{L}_{val}(\theta) - \mathcal{L}_{val}(\theta')$. In words, the agent proposes a synthetic data query $\mathbf{a}$ and the oracle P provides a label $\mathbf{s}'$. The agent's objective is to reduce validation loss with a minimal number of data queries. Most active learning methods take a greedy approach to maximize the model loss reduction after a single data query which corresponds to setting $\beta = 0$.

Pool-based Active Learning is the same as Query Synthesis Active Learning with the only difference being $\mathcal{A} = \mathcal{D}_{pool}$ where $\mathcal{D}_{pool}$ is the initial pool of unlabelled data.

Stream Active Learning is obtained by choosing $\mathcal{S} = \mathcal{X} \times \mathcal{Y}, \mathcal{A} = \{0, 1\}, P(\cdot|\mathbf{s} = (\mathbf{x}, \mathbf{y}), \mathbf{a}) = \omega(\mathbf{x})$ if $\mathbf{a} = 1$ else $\delta(\mathbf{y}_{dum})$, and $c(\bar{\mathbf{s}} = (\mathbf{s}, H, \theta), \mathbf{a}, \bar{\mathbf{s}}' = (\mathbf{s}', H', \theta')) = \mathcal{L}_{val}(\theta) - \mathcal{L}_{val}(\theta')$, where δ is the Dirac-delta function and $\mathbf{y}_{dum}$ is a dummy label that denotes the case when no label is returned by the oracle.

D World model architecture ablation and disentanglement

To evaluate the importance of disentanglement in world model architecture, independently of controller choice, we produce datasets for offline training for each task (excluding peekaboo, since the behavior is dependent on the observer's choices, no policy-independent offline training dataset can be constructed). We then train the world model to convergence. We compare the loss of our disentangled world model to an *entangled* LSTM architecture that instead takes as input and predicts all external agents together. As seen in Figure 5, the disentangled architecture significantly outperforms the entangled ablation.

Intuitively, the disentangled architecture performs better because it ignores spurious correlations between causally-unrelated events in the agent's data stream. Formalizing this intuition and explaining why this is particularly salient in our current environment, in contrast to some other situations (Locatello et al., 2018), is an important future direction. Interestingly, the disentangled architecture shares a key feature with the concept known as Theory of Mind, which involves the ability to predict the behaviors of other

agents as a function of inferred mental states, such as beliefs, desires, and goals Astington et al. (1990); Premack & Woodruff (1978); Wellman (1992). A core, though often unstated, assumption behind Theory of Mind is the agent-centric allocation of computational resources. Our disentangled model builds this in as a key feature, suggesting that at least one possible function of Theory of Mind may be to enable statistical disentangling. This certainly requires considerable follow-up work to substantiate.

E TRAINING DETAILS

As shown in Algorithm **??**, we interleave world model and policy updates while interacting with the environment. Specifically we update the both the world model and Q-network with 10 gradient steps per 40 environment steps. Both model updates begin after the buffer is filled with 1000 samples.

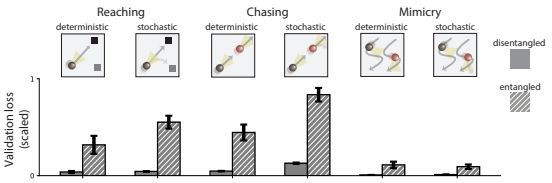


Figure 5: **Asymptotic Model Performance** Final performance of the disentangled world model and entangled ablations.

World Model: We parameterize each component network ω_{θ^k} with a two-layer Long Short-Term Memory (LSTM) network with 256 hidden units if $|I_k| = 1$ i.e., the causal group k contains a single external agent, and 512 if $|I_k| \geq 2$ to ensure that the size of the parameter space scales with the input and output size. All networks are train using Adam with a learning rate of $1e$-4, $\beta_1 = 0.9, \beta_2 = 0.999$ and batch size 256.

The old model is synchronized with the new model weights once after 100 world model updates. This "warm starts" the old model and prevents unreasonable large progress rewards at the start. We use a fixed value of the progress horizon $\gamma = 0.9995$ across all experiments. We found that any $0.9995 \leq \gamma \leq 0.9999$ attains similar results.

Policy Learning: For Q-network Q_ϕ updates we use the DQN algorithm (Mnih et al., 2015) with a discount factor of $\beta = 0.99$, a boostrapping horizon of 200, a buffer size of $2e5$. Same as the world model, we train the Q-network using Adam with a learning rate of $1e$-4, $\beta_1 = 0.9, \beta_2 = 0.999$ and batch size 256. The policy π_ϕ is an ϵ-greedy exploration strategy with respect to Q_ϕ. Specifically, ϵ is linearly decayed from 1.0 to 0.025 at a rate of 0.0001 per environment step.

F VALIDATION CASES

Here we describe validation protocol for each behavior. As data for the world model must be generated by interacting with the environment, what policy to use during validation is an important choice. As some behaviors are "interactive", i.e the external agent dynamics depend on the curious agent's actions, a naive policy that simply stares at the external agent may not elicit the core dynamics underlying the behavior. Thus, we hard-code the policy during validation to elicit the core dynamics for behavior and subsequently measure world model loss on the collected data.

Peekaboo: The validation policy looks at the peekaboo external agent until it hides. The policy then keeps the peekaboo external agent in view so that when the agent "peeks" it immediately hides again. The validation loss measures the world model performance on predicting the dynamics of this peeking behavior which is representative of the core "interactive" nature of peekaboo.

Reaching: At the start of validation, auxiliary objects are spawned at new locations which changes the trajectory of the reaching external agent. The validation policy then stares at the reaching external agent and validation loss is measured on the collected samples. This validation loss measures how well the world model has learned the contingency between the auxiliary object locations and the reaching external agent's movements. For example, a world model that has overfit to the external agent's trajectory for a particular set of auxiliary object locations will fail to generalize when auxiliary objects are spawned at new locations.

Chasing, Mimicry, Periodic, Static, Noise: The validation policy simply stares at the external agents and validation loss is measured on the collected samples.

The validation losses shown in Figure 3a for the Mixture world is an average of the validation losses on the static, periodic, and animate external agents. The random agent is excluded from evaluation as there is virtually no learnable patterns in the behavior and averaging the large world model loss incurred on

the random external agent could occlude the learning performance differences between curiosity signals on the other learnable external agents. For the Noise World, the shown validation losses in Figure 3b represent only the validation loss on the animate external agent.

G NOISE WORLD ATTENTION

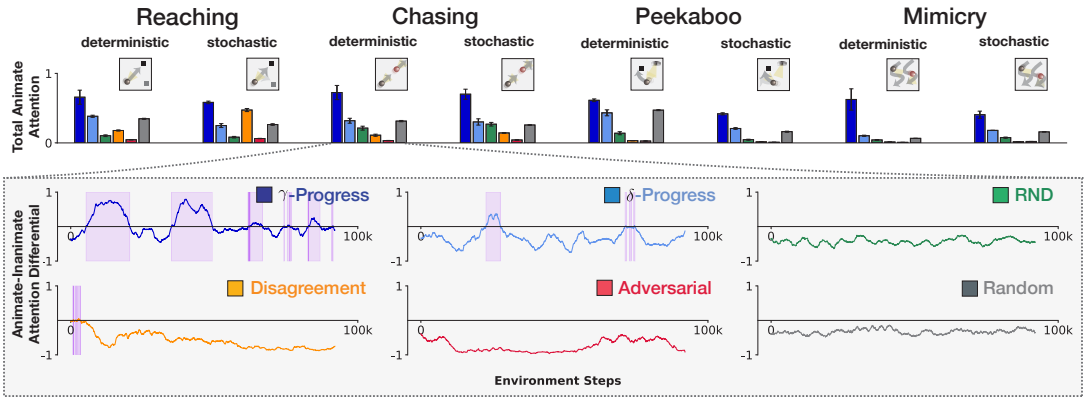


Figure 6: **Attention Patterns in Noise World**. The bar plot shows the total animate attention, which is the ratio between the number of time steps an animate external agent was visible and the number of time steps a noise external agent was visible. The zoom-in box plots show the differences between mean attention to the animate external agents and the mean of attention to the other agents in a 500 step window, with periods of animate preference highlighted in purple. Results are averaged across 5 runs. γ-Progress displays strong animate attention while baselines are either indifferent, e.g δ-Progress, or fixating on white noise, e.g Adversarial.

H FURTHER ATTENTION ANALYSES

Here we provide details of the early indicator analysis (Section **??**) and a regression of what factors (curiosity signal, architecture, external agent behavior) best predict animate/inanimate attention ratios.

H.1 DETAILS OF EARLY INDICATOR ANALYSIS

We look to predict final performance P_{final} of a given agent, which we take to be the average of the final four validation runs. To make the modeling problem simple, we discretize this into a classification task by dividing validation performance into 3 equal-sized classes ("high", "medium", and "low", computed separately for each external agent behavior), intuitively chosen to reflect performance around, at, and below that of random policy.

We consider two predictive models of final performance, one that takes as input early attention of the agent, and the other, early performance. Early performance may be quantified simply: given time T ("diagnostic age") during training, let $P_{\leq T}$ be the vector containing all validation losses measured up to time T. Early attention, however, is very high-dimensional, so we must make a dimensionality-reducing choice in order to tractably model with our modest sample size. Hence, we "bucket" average. Given choice of integer B, let

$$A_{\leq T,B} = (f^{\text{anim}}_{0:\frac{T}{B}}, f^{\text{rand}}_{0:\frac{T}{B}}, f^{\text{anim}}_{\frac{T}{B}:\frac{2T}{B}}, f^{\text{rand}}_{\frac{T}{B}:\frac{2T}{B}}, \dots f^{\text{anim}}_{\frac{(B-1)T}{B}:T}, f^{\text{rand}}_{\frac{(B-1)T}{B}:T}), \tag{13}$$

where $f^{\text{anim}}_{a:b}$ and $f^{\text{rand}}_{a:b}$ are the fraction of the time $t = a$ and $t = b$ spent looking at the animate external agent and random external agents respectively (so $A_{\leq T,B}$ is the attentional trajectory up to time T discretized into B buckets).

Finally, both models must have knowledge of the external agent behavior to which the agent is exposed — we expect this to both have an effect on attention as well as the meaning of early performance and expected final performance as a result. Let χ_{BHR} be the one-hot encoding of which external animate agent behavior is shown.

We then consider models

Table 1: **Attention regression.** Regression model of animate/noisy attention, according to Equation 14. Coefficient values found, and uncorrected p-value for 2-sided t-tests, with significance at the .05 level in bold.

COEFFICIENT	VALUE	P > |T|
CONSTANT	.80	.001
γ-PROGRESS	**2.24**	.000
δ-PROGRESS	.08	.788
RND	-.53	.064
DISAGREEMENT	**-.70**	.014
ADVERSARIAL	**-.79**	.006
CAUSAL ARCHITECTURE	.014	.959
STOCHASTIC REACHING	.14	.493
DETERMINISTIC CHASING	.25	.222
STOCHASTIC CHASING	**.45**	.029
DETERMINISTIC PEEKABOO	-.08	.682
STOCHASTIC PEEKABOO	.02	.920
MIMICRY	**.56**	.006
CAUSAL $\times\gamma$-PROGRESS	-.32	.408
CAUSAL, $\times\delta$-PROGRESS	.06	.868
CAUSAL $\times$ RND	.03	.935
CAUSAL $\times$ DISAGREEMENT	.23	.555
CAUSAL $\times$ ADVERSARIAL	-.09	.813

1. $\text{PERF}_{\leq T}$, which takes as input $P_{\leq T}$ and χ_{BHR}, and
2. $\text{ATT}_{\leq T}$, which takes as input $A_{\leq T,B}$ and χ_{BHR}.

Figure **??**b shows the plot of $\text{PERF}_{\leq T}$ and $\text{ATT}_{\leq T}$ accuracy as T varies. We see that, up to a point, $\text{ATT}_{\leq T}$ makes a better predictor of final performance, and then $\text{PERF}_{\leq T}$ dominates. This confirms the intuition that attention patterns precede performance improvements. Intuitively, early attention predicts performance by being able to predict the sort of curiosity signal the agent is using, which predicts the full timecourse of attention (see H.2), which in turn predicts performance.

H.2 DETERMINANTS OF ATTENTION PATTERN

To gain a finer-grained understanding of what, of the factors we vary (curiosity signal, world model architecture, and stimulus type) drives the attentional behavior of these active learning systems, we perform a linear regression. Specifically, we regress

$$R_{\text{animate/noisy}} = a + b \cdot \chi_{\text{CS}} + c\chi_{\text{causal}} + d \cdot \chi_{\text{BHR}} + \chi_{\text{causal}} * e \cdot \chi_{\text{IM}} + \epsilon \tag{14}$$

Here $R_{\text{animate/noisy}}$ is the ratio of animate to noisy attention, χ_{CS} is a one-hot encoding of curiosity signal (all zeros if random policy), χ_{causal} is an indicator set to 1 if the architecture is causal, χ_{BHR} is a one-hot encoding of animate external agent behavior shown (all zeros if deterministic reaching), and a, b, c, d, e are fixed effects (e measures an interaction effect).

Over 371 individual active learning runs, an ordinary least squares regression achieves an adjusted R^2 of .44. Please see Table 1 for details. We found that γ-Progress receives significant positive weight, while Disagreement and Adversarial receive significant negative weight, with the other curiosity signals having an effect close to that of random policy. In addition, we fail to find a significant effect due to architecture and most external agent behaviors, with two external agent behavior exceptions. In sum, we find that, of the architectural and curiosity signal variations we tested, curiosity signal strongly drives behavior whereas architecture plays an insignificant role.