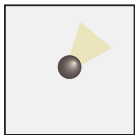
Appendix - Active World Model Learning in Agent-rich Environments with Progress Curiosity

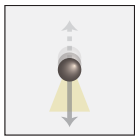
A. External Agent Behaviors

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.

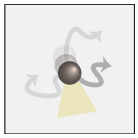
Inanimate behaviors



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

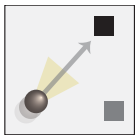


Periodic Inspired by objects exhibiting periodic motion such as fans, flashing lights, and clocks, the *periodic agent* regularly moves back and forth between two specified locations in its quadrant.

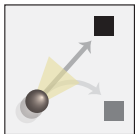


Noise Inspired by random motion in wind, water, and other inanimate elements, the *noise agent* randomly samples a new direction and moves in that direction with a fixed step size while remaining within the boundaries of its quadrant.

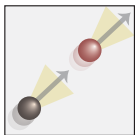
Animate Behaviors



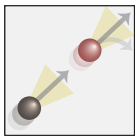
Reaching (deterministic) We often exhibit goal-oriented behavior by interacting with objects. The *reacher agent* approaches each auxiliary object in its quadrant sequentially, such that object positions fully determine its trajectory. Objects periodically shift locations such that predicting agent behavior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



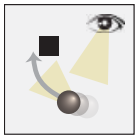
Reaching (stochastic) The order in which the reacher agent visits the objects is stochastic (uniform sampling from the three possible objects). However, once the reacher agent starts moving towards an object, its trajectory for the next few time steps, before it chooses a different object to move to,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



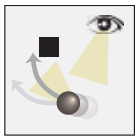
Chasing (deterministic) We often act contingently on the actions of other agents, which in turn depend on our own. In chasing, a *chaser agent* chases a *runner agent*. If the runner is too close to quadrant bounds, it then escapes to one of a few escape locations away from the chaser but within the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



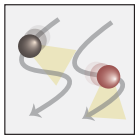
Chasing (stochastic) When the runner agent is too close to the quadrant bounds, it escapes by picking any random location away from the chaser and within the bounds of the quadrant.



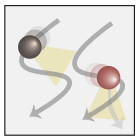
Peekaboo (deterministic) One way of detecting an animate agent is if its motion is contingent on our own. The *peekaboo agent* acts contingently on the curious agent. If the curious agent stares at it, it hides behind an auxiliary object such as a doll. If the curious agent continues to stare, it starts *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. Connections between AWML and Curiosity

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

$$-c(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}(\theta, \mu) - \mathcal{L}(\theta', \mu) \simeq D_{\mathrm{KL}}(p(\theta')||p(\theta)) \tag{7}$$

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 $D_{\mathrm{KL}}(q(\theta')||q(\theta))$ (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).

Adversarial (Stadie et al., 2015; Pathak et al., 2017; Schmidhuber, 2019) curiosity assumes prediction gain is proportional to the current world model loss.

$$-c(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}(\theta, \mu) - \mathcal{L}(\theta', \mu) \simeq -\log \omega_\theta(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \tag{8}$$

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 function ω on inputs $\mathbf{x}$ for which $\omega(\mathbf{x})$ is not a Dirac-delta distribution. This problem is known as the white noise problem (Schmidhuber, 2010).

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

$$-c(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}(\theta, \mu) - \mathcal{L}(\theta', \mu) \simeq \mathrm{Var}(\{\omega_{\theta_j}(\mathbf{s}'|\mathbf{s}, \mathbf{a})\}_{j=1}^N) \tag{9}$$

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., 2018b) methods reward transitions with a low visitation count $\mathcal{N}(s, a, s')$. The prototypical novelty reward is:

$$-c(\bar{\mathbf{s}}, \mathbf{a}, \bar{\mathbf{s}}') = \mathcal{L}(\theta, \mu) - \mathcal{L}(\theta', \mu) \simeq \mathcal{N}(s_t, a_t)^{-1/2} \tag{10}$$

(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.

C. 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 6, 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.

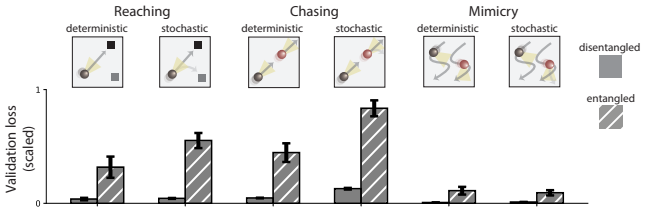


Figure 6. **Asymptotic Model Performance** Final performance of the disentangled world model and entangled ablations.

D. Training Details

As shown in Algorithm 1, 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.

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.

E. 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.

F. Noise World Attention

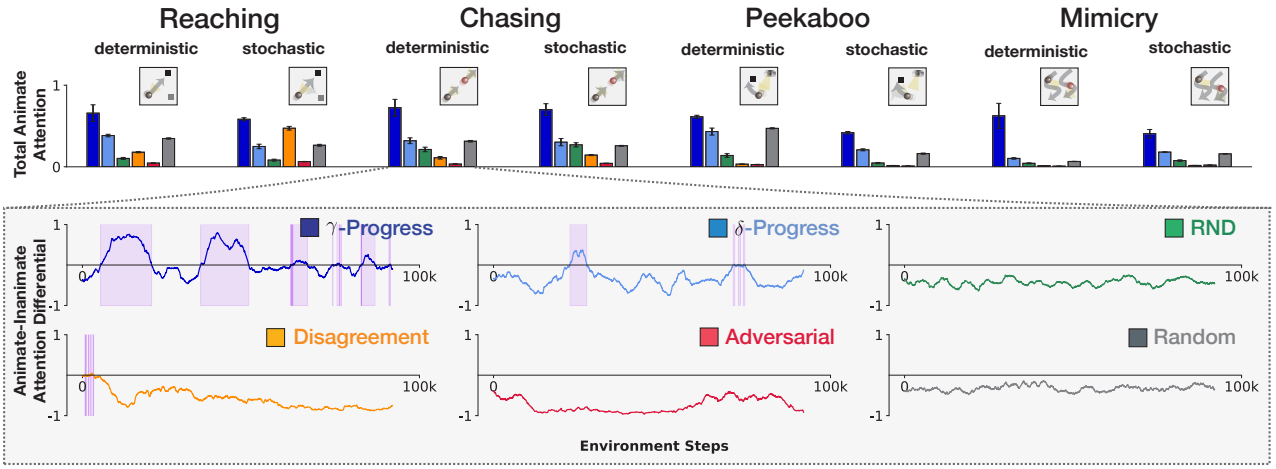


Figure 7. **Attention Patterns in Noise World**. The bar plot shows the total animate attention, which is the ratio between the number of time steps an animate external agent was visible and the number of time steps a noise external agent was visible. The zoom-in box plots show the differences between 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.

G. Further attention analyses

Here we provide details of the early indicator analysis (Section 7) and a regression of what factors (curiosity signal, architecture, external agent behavior) best predict animate/inanimate attention ratios.

G.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

$$A_{\leq T,B} = (f^{\text{anim}}_{0:\frac{T}{B}}, f^{\text{rand}}_{0:\frac{T}{B}}, f^{\text{anim}}_{\frac{T}{B}:\frac{2T}{B}}, f^{\text{rand}}_{\frac{T}{B}:\frac{2T}{B}}, \dots f^{\text{anim}}_{\frac{(B-1)T}{B}:T}, f^{\text{rand}}_{\frac{(B-1)T}{B}:T}), \tag{11}$$

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).

Table 3. **Attention regression.** Regression model of animate/noisy attention, according to Equation 12. 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

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

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 5b 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 G.2), which in turn predicts performance.

G.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

$$R_{\text{animate/noisy}} = a + b \cdot \chi_{\text{CS}} + c\chi_{\text{causal}} + d \cdot \chi_{\text{BHR}} + \chi_{\text{causal}} * e \cdot \chi_{\text{IM}} + \epsilon \tag{12}$$

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 3 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.