DOUBLE TAKE: INTRINSICALLY MOTIVATING AGENTS TO MODEL MANY WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the problem of an agent with prior beliefs over a distribution of possible worlds, who wants to infer the identity of the new world in which it finds itself at runtime. Supposing the agent has some implicit belief about the current world, we show that a surrogate for information gain can be computed by training a model to predict multiple possible outcomes of an action given some observed history in the world, then asking the model how much more likely an outcome becomes after observing it once. The benefit of such a "double take" is that first looks are informative only insofar as they reveal the hidden structure of the environment; by construction, they cannot help predict noisy TVs. However, assuming a black-box environment, we cannot sample alternate outcomes conditioned only on an agent's observations if the environment includes noisy hidden state. To get around this, we instead perform an entire "second take" of agent interaction in the same world, and train a model to better predict the first take using information from the second. We can then simulate the effect of an instantaneous double-take by replaying the first look to the model as a transition within a supposed first take, and having it make new predictions using the shared-world structure this disguised observation confers. We use procedurally-generated reinforcement learning environments to model the problem of information gathering over many worlds, measuring the amount of information gathered about the world by the resultant agents relative to an ideal Bayesian observer. Our methods outperform intrinsic-RL baselines in terms of world-information-gathering, and often discover behavior correlating with high environment reward. They may also provide a general recipe for collecting relevant data for training interactive next-frame-prediction world models.

1 INTRODUCTION

Curiosity is a force that intrinsically motivates agents to explore new environments and learn how they work (Schmidhuber, 1991). A classic formalization of curiosity is to maximize *information gain* about the environment's hidden dynamics (Storck et al., 1995; Yi et al., 2011). This involves rewarding a new observation by comparing how surprising it is at first glance, and how surprising it would be a second time were it to have been used to update a belief over worlds (eq. (2)). But maintaining an explicit representation of this latter belief is intractable in sufficiently rich environments (Houthooft et al., 2016). Meanwhile, modern neural world models, from Genie (Bruce et al., 2024) to Sora (Brooks et al., 2024) to BabyZWM (Aw et al., 2026), have no such explicit posterior, and yet are capable of adapting to variable world dynamics at runtime by learning in-context given recent history. This would seem to suggest that these fundamentally straightforward next-frame-prediction models maintain and update some implicit beliefs about the current world.

In this paper, we show how the implicit beliefs of neural world models can be leveraged for curiosity and intrinsic motivation. Assuming a simulator of possible worlds, we show how to *metalearn* a neural world-modeling agent who, when faced with some new world, can efficiently learn to model it at runtime by harvesting information about its latent dynamics. This is achieved by using the world model's in-context predictions as surrogates for information gain, then training a policy to take actions that are informative according to this measure.[1]

[1] Code and demo videos are available at this anonymized link.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

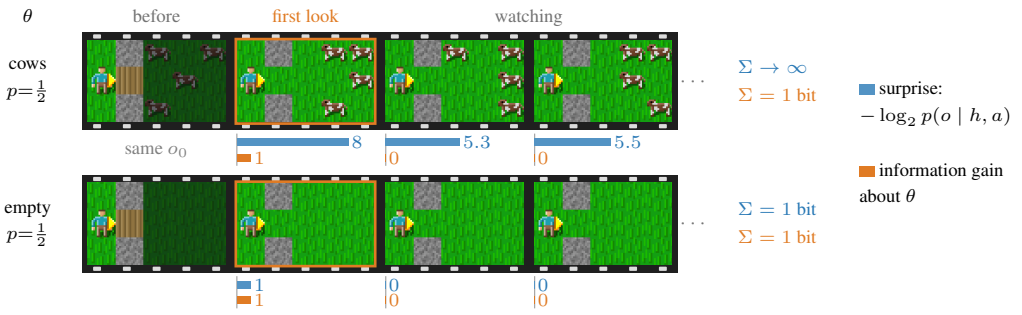


Figure 1: **Information gain ignores noisy TVs.** Consider a TV that is known to be either tuned to static or broken. When we attempt to turn the TV on, we gain one bit of information (reducing our epistemic uncertainty about the world). This is true even when the TV is working but tuned to static, even though each new frame of noise is difficult to predict (providing endless aleatoric uncertainty).

Our surrogate for information gain relies on what we call *double-takes:* given a history of observations-actions pairs, our model learns to predict *two* possible next observations, one with, and one without, some additional independent history drawn from the same world. This allows the model to distinguish epistemic uncertainty about the world's hidden identity (which may be resolved with reference to the supporting history) from aleatoric uncertainty resulting from any irreducible stochasticity within it (which may not), where these confounds would otherwise be folded together in a standard next-frame predicting neural world model.

The practical result is a scalable method for metalearning curiosity that builds on the contemporary world-model AI stack: simple autoregressive backbones exhibiting in-context learning, metalearned on synthetic data. We also prove that any similar approach building on this stack *must* use a simulator: no next-frame predictor can serve as a surrogate of information gain *without* privileged structure only available in simulation, intuitively due to a certain non-identifiability of world dynamics. Together this shows our assumptions are theoretically necessary, and the experiments show them to be practically sufficient.

2 Related Work

Prediction-based curiosity and the noisy TV. A dominant approach to intrinsic motivation in learning agents is to reward the agent for its ability to surprise—i.e. elicit error from—some model of the environment which learns along with the agent (Stadie et al., 2015; Guo et al., 2022). But noise is inherently hard to model, and so intrinsic learners risk conflating epistemic and aleatoric uncertainty; that is, about how the world works on one hand and about inherently unpredictable outcomes of stochastic processes within it on the other (i.e. "noisy TVs", Figure 1). As such, curious world-modeling agents risk being drawn to irreducibly stochastic fountains of information.

Prediction in latent space. Rather than predicting raw observations, many curiosity methods measure surprise in an embedding space, and the choice of this space largely determines what the agent finds interesting. **BYOL-Explore** (Guo et al., 2022) is a representative example. Following the self-supervised Bootstrap Your Own Latents (BYOL) objective (Grill et al., 2020), it trains a network to predict the embeddings of future observations, with target embeddings supplied by a slowly-updated copy of the same network. Because the target is learned, the representation can discard details that are hard to predict, but nothing explicitly separates noise from structure, and BYOL-Explore remains vulnerable to stochastic distractors (Jarrett et al., 2022). Other methods make different choices of latent space: they use fixed random embeddings, or learn one through inverse dynamics (see below).

Random Network Distillation (RND) (Burda et al., 2019) trains an intrinsic agent capable of "getting used to" the TV. In brief, a student model is given observations emitted during world-interaction and asked to predict the corresponding low-dimensional embedding that is produced when the same observation is fed to a frozen, randomly-initialized target network. The agent is rewarded by this student's prediction error. States that are visited frequently will be memorized by the student, depleting intrinsic reward for the agent and forcing it to move onto something more surprising. But the

student must still memorize a space of embeddings proportional to the space of all frames of noise, even though that does nothing for the agent's ability to understand the world.

To discern novelty that has some bearing on the agent, other intrinsic learners rely on *inverse dynamics models* (Agrawal et al., 2016). These models embed successor observations o_t and o_{t+1} into latents e_t and e_{t+1}, and are trained both to predict e_{t+1} from e_t, and the agent's action a from e_t and e_{t+1}. In this way the student model is incentivized to ignore elements of the observation that do not inform the agent's actions. In the **Intrinsic Curiosity Module (ICM**) (Pathak et al., 2017), the agent is rewarded by the forward prediction error of such a model in the latent space. But not everything that impacts or is impacted by the player's actions corresponds to meaningful information about the world. An agent in a ballpit of randomly moving orbs may collide with these orbs extensively, so that a student of inverse dynamics should track the locations of these orbs to understand the agent's motion within the pit. But the student will inevitably fail to predict the irreducibly random orbs, incentivizing the agent to stay in the pit in spite of its trivial underlying dynamics.

Exploration via Elliptical Episodic Bonuses (E3B) (Henaff et al., 2022) learns a similar inverse dynamics student model, but instead uses its latent space to compute the similarity of observations witnessed over the course of an episode. Agents are rewarded for taking actions that maximize novelty in the latent space, encouraging them to visit during each episode as many states as can be recognized as distinct by the student. Due to the noisy-player-impact pathology described above, this could include a vast number of diverse configurations inside the trivial ballpit.

Henaff et al. (2023) further differentiates between *global* intrinsic RL bonuses (e.g. RND, ICM), which reward novelty experienced over the course of learning during training, and *episodic* bonuses, (including E3B and other count-based approaches (Zhang et al., 2021)), which reward novelty over the course of an episode. This motivates hybrid approaches like **E3B** $\times$ **RND** which combines (in this case multiplies) one of each such bonus. Double Take is an episodic bonus, asking "does this reveal something new about the environment relative to the rest of this episode?" But it is more specifically a procedural bonus: this intra-episode novelty is determined by the structure of the space of possible worlds.

Other approaches to curiosity involve measuring disagreement across an ensemble of forward models (Pathak et al., 2019; Shyam et al., 2019; Sekar et al., 2020), and explicitly estimating aleatoric noise and subtracting it (Mavor-Parker et al., 2022). Curiosity in Hindsight (i.e. **BYOL-Hindsight**) (Jarrett et al., 2022) conditions the forward model on a hindsight representation of the realized future, which is trained to maximize reconstruction accuracy while minimizing mutual information with the current history. However, this hindsight can learn to smuggle world-specific information from the future when its reveal is independent of prior events. Double Take similarly provides a kind of "hindsight" to the model by conditioning it on the true next frame, but by using paired, independent rollouts to it learns to explicitly discern information about shared world structure.

Johnson et al. (2024)'s *Experts Don't Cheat* employs a sleight of hand similar to Double Take to show that models are more likely to "cheat" when shown an answer at runtime if they are themselves uncertain about the answer. A Double Take agent seen through this lens is one who recognizes that there is something to learn when it catches itself cheating by plagiarizing from a peek into the future. Double Take differs in that its scoring-time cheating/double-take is out-of-distribution, whereas Johnson et al. are able to train on independent responses to natural language queries, where sometimes identical responses are more likely to be sampled during paired training.

3 Background and Problem Statement

We assume no external reward, and seek to define intrinsic rewards that serve as proxies of the amount of information an agent gains about how the world works. We formalize this as an agent interacting with a *controlled Hidden Markov Model* (a POMDP without reward), with unknown parameters θ. The agent is motivated to minimize its uncertainty over θ.

Controlled Hidden Markov Models. A controlled hidden Markov model (CHMM) is a tuple $\langle \mathcal{O}, \mathcal{A}, \mathcal{Z}, \theta \rangle$ of observation space $\mathcal{O}$, action space $\mathcal{A}$, hidden states $\mathcal{Z}$, and parameters θ. Parameters θ define distributions $p(o \mid z, \theta)$ for emissions, $p(z' \mid z, a, \theta)$ for transitions, and $p(z \mid \theta)$ an initial state prior, where $o \in \mathcal{O}$, $a \in \mathcal{A}$, $z \in \mathcal{Z}$.

Histories. We write h for a history of observations and actions, where each history is a sequence beginning with an observation and alternating between actions and observations. Each CHMM defines a distribution $p(o \mid h, a, \theta)$ over next observations given history h and action a whenever h ends in an observation, given by marginalizing the hidden states.

Dynamics priors, posterior, posterior predictive. Curiosity requires quantifying belief—how much uncertainty remains about how the world works, and will an experiment, plan, or action reduce that uncertainty? We therefore assume a prior $p(\theta)$ over the parameters that govern the world's dynamics, which then defines the posterior $p(\theta \mid h)$ and the posterior predictive $p(o \mid h, a) = \sum_\theta p(\theta \mid h)p(o \mid h, a, \theta)$.

Information Gain. A classic intrinsic reward for taking action a in history h is the reduction in world model uncertainty after observing the resulting observation o, or *information gain* (IG),

$$\mathrm{IG}(o; h, a) = \mathbb{H}\left[\theta \mid h\right] - \mathbb{H}\left[\theta \mid h, a, o\right] \tag{1}$$

$$= -\log \underbrace{p(o \mid h, a)}_{\text{“first take”}} - \mathbb{E}_{\theta \sim p(\cdot \mid h, a, o)}\left[-\log \underbrace{p(o \mid h, a, \theta)}_{\text{“double-take”}}\right] \tag{2}$$

which eq. (2) rewrites in terms of next-observation predictions: infogain is equivalent to the surprise-gap of the next observation under the prior and posterior before/after observing o. In other words we gain information from initially surprising observations (low $p(o \mid h, a)$), but conditioned upon having already seen it, a second "double-take" of that same observation yields little surprise (high $p(o \mid h, a, \theta)$ with θ drawn conditioned on seeing o).

To see why information gain is a desirable reward, consider the "noisy TV" problem. An agent that simply chases surprise ($-\log p(o \mid h, a)$) is trapped by the static on a noisy TV because it yields endless surprise but no information.

Witnessing the static on a noisy TV, however, does not help predict a "double-take" of the static: Every time the noisy TV is observed, its contents remain random. In contrast, consider an agent uncertain whether turning on the TV produces static or a blank screen. Turning on the TV collapses this epistemic uncertainty, and whatever the agent observed (noise or blank screen) is made less surprising. Therefore, if a neural network learned not merely to predict the next observation, but also to predict the next observation conditioned on a counterfactual past history that coincidentally includes the latest observation, then this would tease apart the noisy TV from the uncertain TV: Predicting a noisy TV does not benefit from a counterfactual observation of static, while an uncertain TV is easier to predict given a counterfactual observation.

Autoregressive neural world models. An autoregressive neural world model is a neural network that predicts the next observation given a history of past observations (and typically, also given actions). This dispenses with explicit hidden state, and dispenses with explicit representation of world dynamics. Such models operate purely in terms of predictive quantities on observed data, i.e. $p(o \mid h, a)$, rather than operate over latent states z and hidden dynamics θ. Such models also support simple and scalable inference and training because autoregressive neural world models make no reference to latent random variables (θ, z). Yet information gain (IG) would appear to require reasoning about exactly these latent random variables, naively precluding information gain as an intrinsic motivator when using these models.

In-context posterior predictive quantities. Autoregressive sequence models learn in-context to approximate the Bayesian posterior predictive (Müller et al., 2022). If a neural world model is trained on rollouts from a fixed CHMM with parameters θ, their next-observation predictions approximate $p(o \mid h, a, \theta)$. If trained on rollouts from a distribution of CHMM's with parameters drawn from $p(\theta)$, their next-observation predictions approximate $p(o \mid h, a)$.

Insufficiency of posterior predictive distributions for information gain. Could an autoregressive world model, by approximating the posterior predictive, be used to maximize information gain? We answer this question in the negative by showing that any dataset of rollouts could be generated by different generative processes that dramatically differ in their information gain. As the posterior predictive $p(o \mid h, a)$ depends only on the distribution of rollouts, this renders information gain under-determined.

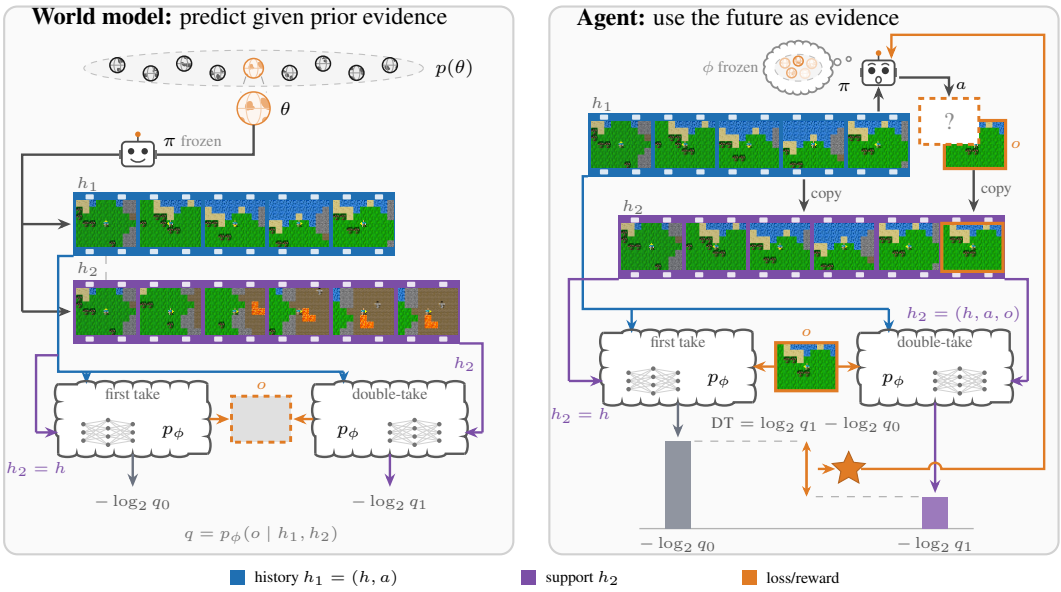


Figure 2: **Learning to simulate a double-take.** To train our world model, we sample an environment θ, apply to it the agent policy π to generate two rollouts h_1 and h_2, and train the world model to predict the first using the second as prior evidence. The policy is rewarded for eliciting states that surprise the world model in terms of its simulated double-take, i.e. that yield futures which cause the world model to refine its prediction when disguised as prior evidence.

Concretely, suppose we have a prior over parameters $p(\theta)$, which induces the posterior predictive $p(o \mid h, a) = \sum_\theta p(\theta \mid h)p(o \mid h, a, \theta)$. Construct a CHMM indexed by ψ whose dynamics are independent of ψ and always equal to the posterior predictive induced by $p(\theta)$, i.e. $p(o \mid h, a, \psi) = \sum_\theta p(\theta \mid h)p(o \mid h, a, \theta)$. For any prior $p(\psi)$ the induced posterior predictive is identical to the posterior predictive induced by $p(\theta)$, therefore $p(\psi)$ gives the exact same training distribution as $p(\theta)$. By inspection, any h, a, o the information gain w.r.t. ψ is 0, but the information gain w.r.t. θ will generally be non-zero. Therefore, the distribution of histories and resulting posterior predictives cannot distinguish a scenario with positive information gain from a symmetric construction with zero information gain everywhere. Thus, a raw stream of interaction histories over worlds is insufficient for collecting information about θ: we need data with some additional structure from the environment generator (e.g. histories paired according to world-provenance).

4 Methods

Consider an agent metalearning to gain information over a distribution of worlds (distribution of θ's). In each world its actions should disambiguate the actual world from other possible worlds, for example by eliciting transitions that can occur in one world but not others. In section 3 we established that this can be captured by the information gain, which is computed as a difference between a prior prediction term (the "first take") and a prediction term conditioned on the posterior (the "double-take"). The double-take term is very tricky to compute directly, since it requires us to marginalize out the posterior distribution over worlds $p(\theta \mid h, a, o)$. In this section we are going to show how to compute proxies to information gain without having to worry about this term.

The Double-Take substitution trick We build toward our objective by first showing a simpler approach to reducing information gain to posterior predictive quantities. Notice that the double-take term can be thought of as a retroactive prediction: how likely is observation o, given my updated belief about the world θ after seeing o itself? To make this intuition concrete, suppose we have

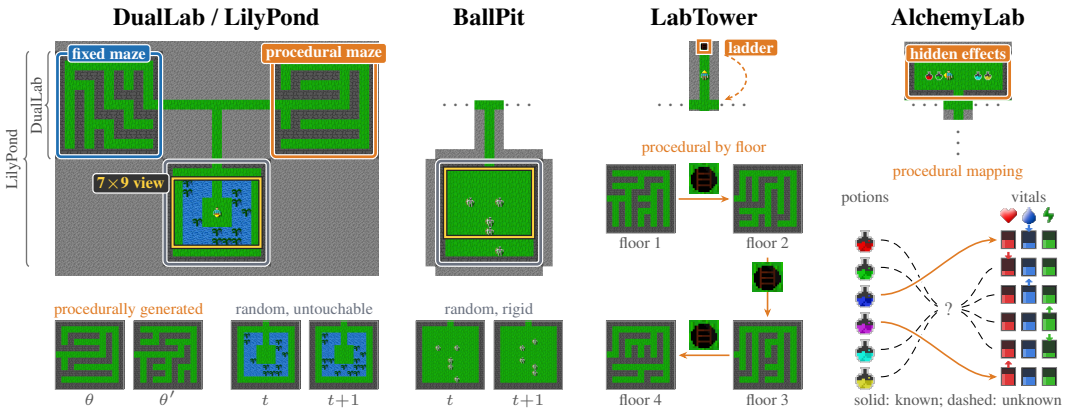


Figure 3: **The Lab suite probes curious agents' information-gathering abilities.** DualLab offers one fixed and one procedural maze room. LilyPond adds a noisy TV distractor-room. BallPit extends this noise to respond to agent interaction. LabTower adds an infinite series of procedural floors to test information-gathering over long horizons. AlchemyLab adds a hidden potion-mapping to test the effect of competing information sources.

independent samples $o, o' \sim p(\cdot \mid h, a, \theta)$, then:

$$\log p(o \mid h, a, o') = \log \mathbb{E}_{\theta \sim p(\cdot \mid h,a,o')} \left[p(o \mid h, a, \theta) \right] \geq \mathbb{E}_{\theta \sim p(\cdot \mid h,a,o')} \left[\log p(o \mid h, a, \theta) \right] \tag{3}$$

Thus learning the posterior predictive $p(o \mid h, a, o')$, and at test time setting $o' = o$, we recover an upper bound on the Double Take term in eq. (2).

Unfortunately, in stochastic environments with partial observability, generating multiple samples from $p(\cdot \mid h, a, \theta)$ is generally intractable, as it requires marginalizing out all possible hidden states z that are consistent with the observations so far. Since the hidden states are not fixed by the world θ and agent actions due to stochasticity, the integral can quickly become intractable even with relatively simple dynamics. This becomes important for environments like Craftax, where both stochasticity and partial observations are core aspects to its appeal and challenge.

Information gain via double-counted histories Since we do not have access to an independent sample o', we take an alternative approach. Instead, we consider two conditionally independent histories $h_1, h_2 \sim p(\cdot \mid \theta, \pi)$, where π is the agent's current policy. We can compute a similar quantity:

$$p(o \mid h_1 = (h, a), h_2 = (h, a, o)) = \mathbb{E}_{\theta \sim p(\cdot \mid h_1 = (h,a), h_2 = (h,a,o))} \left[p(o \mid h, a, \theta) \right] \tag{4}$$

$$\approx \mathbb{E}_{\theta \sim p(\cdot \mid h,a,o)} \left[p(o \mid h, a, \theta) \right], \tag{5}$$

where we are using the posterior that double-counts the history $p(\theta \mid h, a, h' = (h, a, o))$ as a proxy to $p(\theta \mid h, a, o)$. To offset this double counting, we also double count the history in the first take, resulting in the Double-take infogain:

$$\mathrm{DT}\,(o; h, a) = -\log \underbrace{p(o \mid h_1 = (h, a), h_2 = h)}_{\text{first take}} + \log \underbrace{p(o \mid h_1 = (h, a), h_2 = (h, a, o))}_{\text{double-take}} \tag{6}$$

$$= -\log q(o \mid h_2 = h) + \log q(o \mid h_2 = (h, a, o)) \tag{7}$$

$$= D_2 \left(q(\theta \mid h_2 = (h, a, o)) \,||\, q(\theta \mid h_2 = h) \right), \tag{8}$$

where D_2 is the Rényi divergence of order 2, and $q(\cdot \mid h_2) \triangleq p(\cdot \mid h_1 = (h, a), h_2)$ (see section A). Importantly, as both KL and Rényi divergences are nonnegative and vanish iff their arguments are equal, it follows that our reward function is zero iff information gain is zero, and positive iff information gain is positive.

5 Experiments

To understand agents' ability to collect relevant information about the world, we design several toy environments based on the engine and RL agent interface underlying Craftax (Matthews et al.,

Table 1: **Double Take collects the most information about the world across the Lab suite.** Mean bits collected ± one standard error over 5 training seeds (reset seeds for Uniform random). Oracle PPO is rewarded with the true information each step reveals. Bold: best non-oracle method.

Method Max. bits	DualLab 16	LilyPond 16	BallPit 16	LabTower ∞	AlchemyLab 25.5
Uniform random	13.3 ± 0.3	9.8 ± 0.4	9.4 ± 0.4	11.7 ± 0.7	16.8 ± 0.5
E3B	$\mathbf{16.0 \pm 0.0}$	13.0 ± 3.0	2.2 ± 0.8	1.5 ± 0.6	13.4 ± 3.8
E3B × RND	12.8 ± 3.2	13.0 ± 3.0	7.6 ± 3.4	7.8 ± 3.5	15.5 ± 4.1
RND	15.9 ± 0.0	6.1 ± 1.4	8.0 ± 1.8	10.6 ± 2.4	17.8 ± 1.4
ICM	15.7 ± 0.3	7.3 ± 3.0	3.7 ± 2.2	3.1 ± 2.0	6.8 ± 4.1
BYOL-Explore	15.8 ± 0.1	15.3 ± 0.2	6.4 ± 3.6	11.5 ± 2.8	3.0 ± 3.0
BYOL-Hindsight	15.9 ± 0.0	14.1 ± 0.8	12.4 ± 2.9	9.8 ± 3.6	9.6 ± 3.9
Double Take (ours)	$\mathbf{16.0 \pm 0.0}$	$\mathbf{16.0 \pm 0.0}$	$\mathbf{16.0 \pm 0.0}$	$\mathbf{1{,}066.6 \pm 182.6}$	$\mathbf{24.1 \pm 0.2}$
Oracle PPO	16.0 ± 0.0	16.0 ± 0.0	16.0 ± 0.0	$1{,}248.3 \pm 6.3$	25.5 ± 0.0

2024). We then measure or approximate gathered world-information using our knowledge of the environment generator (details in Appendix B).

Craftax-classic (Matthews et al., 2024) is a GPU-accelerated version of Crafter (Hafner, 2022), which is itself a demake of *Minecraft* (Persson & Bergensten, 2011) featuring a procedurally-generated 64×64-cell gridworld, where the agent gathers resources, crafts tools with increasingly rare materials, and defends against hostile NPCs. Craftax-full, drawing on the mechanics of *NetHack* (Stephenson, 1974; Küttler et al., 2020) adds eight dungeons below the overworld, each with distinct layouts, enemies, and resources. Here, a curious agent should keep pushing downward rather than lingering on floors it has already sufficiently explored, and drink potions to reveal their hidden effects wherever possible. Craftax computes an achievement score based on player behavior, assigning points for significant in-game activies like resource collection, combat, and crafting items.

Among our suite of procedural information-gathering environments (Figure 3), DualLab sees the player spawn in the middle of a corridor with two maze rooms at either end. The left maze is fixed during environment design, while the right is procedurally-generated. We expect a curious world modeling agent to metalearn to learn the procedural maze at each reset, illuminating it while ignoring the fixed maze at runtime. The DualLab's total area is small, and on its surface does not seem to present a particularly difficult exploration challenge, but it helps surface the nature of our objective.

LilyPond adds to DualLab the Craftax analog of a noisy TV, namely a room with an observation deck that overlooks a small pond whereupon lilypads flit in and out of existence at various positions according to a stochastic cellular automaton-style update rule. An agent chasing pure prediction error should be mesmerized by these inherently unpredictable dynamics, despite their underlying triviality. BallPit introduces a noisy TV for the embodied by adding a room of Non-Player Characters (NPCs) that move randomly and can collide with the player. An ideal agent should learn the fixed underlying behavior of these NPCs. LabTower adds a ladder in the north corridor via which the player can access the next "floor" of the environment. Each floor is the same as the last save for a fresh procedurally-generated labyrinth in the east room, so θ is determined by the sequences of these mazes.

AlchemyLab replaces the north ladder with an alchemy room filled with a set of potions each which are mapped to a set of effects at runtime. An agent seeking to maximize information gain should therefore reveal the full procedural maze and taste all but one of the mystery potions. Since the mechanics of potions and their effects are present nowhere else in the environment and can only each be consumed once per episode, though overall information gathering task becomes sparse and asymmetric.

CraftCamp acts as a miniature, bootcamp-style take on full Craftax. It involves a procedural layout of 9 distinct rooms. An information-gathering agent should explore all of the rooms, removing destructable doorways of progressively stronger material (stone, iron, and diamond), requiring progressively more complex crafting, along the way. A chest with the full flight of potions is provided mid-way through the course.

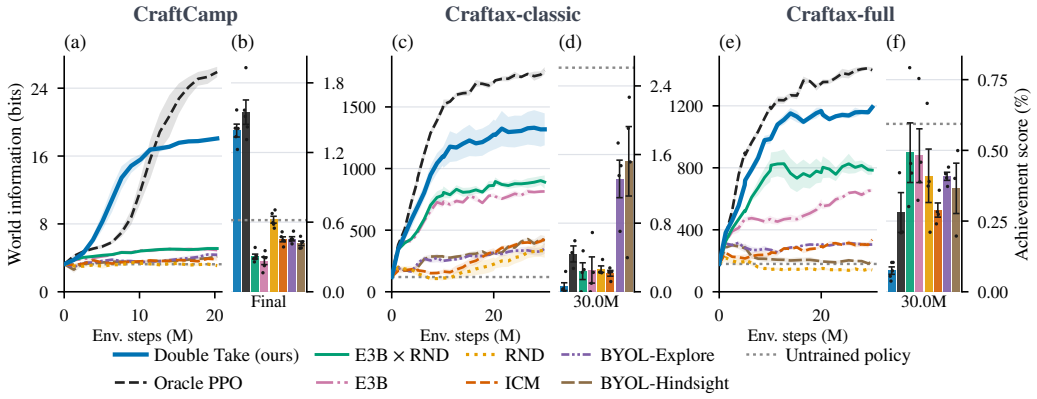
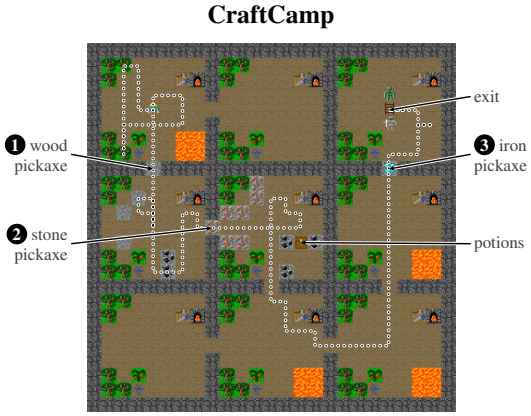


Figure 5: **Double-Take crafts to explore CraftCamp, and efficiently maps the overworld in Craftax.** Performance of Double Take and baselines in CraftCamp, Craftax-classic, and Craftax-full. Double Take collects more procedural world information than baselines (a, b, c). In CraftCamp, it leads in Craftax-native achievement score, because exploration directly requires crafting and mining.

6 Results

Performance in terms of realized information gain across the Lab-suite is reported in Table 1. Most methods do well in the DualLab. Even a uniform random policy reveals $> 75\%$ of the procedural maze. Double Take is the only agent to focus almost exclusively on the procedural room at runtime, because the fixed room is uninformative about the world. Most baselines are robust to the visual noise in LilyPond. For, E3B, for example, it adds only the small path and observatory to the purview of the inverse-dynamics model. RND has to memorize all the pond configurations before moving onto other exploration, but it sometimes manages. BallPit, on the other hand, provides an explosive number of unique states for distinction by both types of baseline student models. Here, Double Take becomes the only approach to still match the Oracle PPO agent's performance: the first look learns to model the distribution of random monster movement well, and it can do no better on the double-take, so the agent stops being rewarded for spending any time in the monster room during new episodes.



Pickaxes unlock the indicated doorways.

Figure 4: CraftCamp miniaturizes the crafting progression of Craftax in a procedural 9-room layout through which the player must mine past increasingly tough rock.

In LabTower, Double Take is the only model to escape the noisy monster room and descend to subsequent floors. It learns to do so systematically, revealing the equivalent of upwards of 70 procedural labyrinths on average over the course of an episode. Often, it takes a relatively greedy approach, only peeking at the threshold of the procedural labyrinth room to uncover the most easily-accessible information about the labyrinth, before taking the ladder down to the next floor. DoubleTake leads information collection in AlchemyLab, largely by learning to harvest information from the maze; its model has difficulty modeling (and thus rewarding) the world-persistent nature of potion-effects, perhaps owing to their scarce, one-shot nature.

At CraftCamp, Double Take accesses more distinct rooms and collects more terrain information than baselines, crafting tools to unlock gates along the way (Figure 5). It initially outpaces the oracle

PPO agent, which is trained directly on the objective, because its WM continues to reward already-observed tiles while it is still learning to remember them, leading to fruitful exploration of early rooms and fortuitous crafting. The strongest baseline (E3B×RND), meanwhile, is liable to spend the bulk of its time locked in the first room, repeatedly (dis)assembling the crafting tables within it to produce a variety of configurations, while sometimes peeking at the random guardians when these are in the adjacent room. Double Take progresses far along the course, visiting 7.6 rooms on average during evaluation against the Oracle's 6.7 and E3B×RND's 1.6; and reaching the penultimate room (before the diamond gate) in 91% of episodes (against 79% and 0%, respectively) (Table 3). The Oracle's lead in world-information harvesting comes entirely from its dutiful potion-consumption, which DT tends to forego in favor of novel terrain.

In Craftax-classic and Craftax-full, Double Take covers substantially more procedural ground than other intrinsic baselines (Figure 5). DT agents will often follow along shorelines and the edges of stone deposits to systematically in-context-learn as much of the map as possible. In doing so, they learn to neglect the crafts in favor of spatial exploration and basic survival with e.g. less wood-harvesting and table-placement than during early learning. In both full and classic, untrained policies earn the highest achievement scores, indicating that baselines have not traded information-gathering for other meaningfully complex behavior, but rather fall victim to noisy distractors in the environment (i.e. NPC movement) of those of their own construction (e.g. object (re)placement across the map). E3B, for example, tends to follow random cows and litter the map with objects.

Procedural pretraining and transfer. As a further test of the capacity to learn general exploration strategies, we pre-train Double Take and baselines on a bank of $4{,}095$ procedurally-generated maze layouts (of sizes 15×15, 21×21, and 27×27), then evaluate performance of the resultant policies on a Craftax-style port of Minihack's Labyrinth-Big (Samvelyan et al., 2021), a large fixed single-corridor labyrinth that proves surprisingly challenging for intrinsically-motivated agents when used as a direct training target (Henaff et al., 2022), along with 9 out-of-distribution maze families each at 9 different spatial sizes (Figure 6). These environments are designed to test transfer across various unseen maze geometries and sizes.

Despite pretraining on relatively small mazes, Double Take manages to transfer to size-201×201 maps without fine-tuning. Double Take achieved the highest final mean information reveal in all 9 families, revealing 46.4% of available information on average compared to 36.9% for E3B×RND, the strongest baseline (Table 2). Crucially, DT's lead increases with the size of OOD mazes (Figure 7). This suggests that it is learning to model novel mazes in-context where baselines may be memorizing fixed sets of states or transitions seen during training. Double Take also was also the most efficient explorer, taking the least number of mean steps ($7{,}113$ across 256 trials compared to $9{,}256$ for E3B×RND) to reach the goal state for Labyrinth-Big (Figure 8. Here as in CraftCamp, we see how Double Take can effectively take advantage of the structure of a well-designed procedural environment generator to achieve downstream goals.

7 Conclusion

Despite first appearances (i.e. their lack of posterior), modern next-frame-prediction neural world models can be used to compute a surrogate of information gain. All it takes is a black-box simulator that allows sampling a pair of rollouts from the same world. We then train the world model not just to predict the result of an action given a history, but given this pair of histories. To train an information-seeking agent, we freeze the world model, pretend that the real next state came from a prior history, and ask the model how much more likely it becomes as a result. By tapping into information gain, our Double Take agents can look past myriad noisy TVs and directly seek to illuminate the nature of the world, where other intrinsic agents cannot discern procedural differences from global or episodic novelty.

A guiding intuition in scaling reinforcement learning is to address overfitting and runtime brittleness by training on vast distributions of procedurally-generated environments (Risi & Togelius, 2020). Double Take agents are particularly well-suited to such a regime because they are explicitly incentivized to explore the structure of these procedural generators. By the same token, they provide a natural means of generating relevant synthetic for training world models capable of inferring individual world dynamics at runtime.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

Khai Loong Aw, Klemen Kotar, Wanhee Lee, Seungwoo Kim, Khaled Jedoui, Rahul Venkatesh, Lilian Naing Chen, Michael C. Frank, and Daniel L. K. Yamins. Zero-shot world models are developmentally efficient learners, 2026. URL https://arxiv.org/abs/2604.10333.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. OpenAI technical report, 2024.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*. PMLR, 2024.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

Zhaohan Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *Advances in neural information processing systems*, 35: 31855–31870, 2022.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1W0z96MFEoH.

Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via elliptical episodic bonuses. *Advances in Neural Information Processing Systems*, 35:37631–37646, 2022.

Mikael Henaff, Minqi Jiang, and Roberta Raileanu. A study of global and episodic bonuses for exploration in contextual mdps. In *International Conference on Machine Learning*, pp. 12972–12999. PMLR, 2023.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Daniel Jarrett, Corentin Tallec, Florent Altché, Thomas Mesnard, Rémi Munos, and Michal Valko. Curiosity in hindsight: Intrinsic exploration in stochastic environments. *arXiv preprint arXiv:2211.10515*, 2022.

Daniel D. Johnson, Daniel Tarlow, David Duvenaud, and Chris J. Maddison. Experts don't cheat: learning what you don't know by predicting pairs. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack learning environment. *Advances in Neural Information Processing Systems*, 33:7671–7684, 2020.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Jackson, Samuel Coward, and Jakob Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2024.

Augustine Mavor-Parker, Kimberly Young, Caswell Barry, and Lewis Griffin. How to stay curious while avoiding noisy tvs using aleatoric uncertainty estimation. In *International conference on machine learning*, pp. 15220–15240. PMLR, 2022.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, pp. 5062–5071. PMLR, 2019.

Markus Persson and Jens Bergensten. Minecraft, 2011. Video game. PC.

Sebastian Risi and Julian Togelius. Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence*, 2(8):428–436, 2020.

Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Kuttler, Edward Grefenstette, and Tim Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=skFwlyefkWJ.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. 1991. URL https://api.semanticscholar.org/CorpusID:18060048.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In *International Conference on Machine Learning*, pp. 5779–5788. PMLR, 2019.

Bradly C. Stadie, Sergey Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *ArXiv*, abs/1507.00814, 2015. URL https://api.semanticscholar.org/CorpusID:10296902.

Mike Stephenson. Nethack, 1974. Video game. PC.

Jan Storck, Sepp Hochreiter, Jürgen Schmidhuber, et al. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pp. 159–164, 1995.

Sun Yi, F Gomez, and J Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *Proc. Fourth Conference on Artificial General Intelligence (AGI), Google, Mountain View, CA*, 2011.

Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Noveld: A simple yet effective exploration criterion. *Advances in Neural Information Processing Systems*, 34:25217–25230, 2021.

A Derivation of the double-take Rényi divergence.

Let h_1 and h_2 denote the query history and support history, respectively. We assume that the query and support history trajectories are generated independently conditioned on the shared world parameter θ under policy π.

For a fixed history h, action a, and observation o, recall the shorthand

$$q(\theta \mid h_2) \triangleq p(\theta \mid h_1 = (h, a), h_2),$$

and in particular

$$q_0(\theta) \triangleq q(\theta \mid h_2 = h), \qquad q_1(\theta) \triangleq q(\theta \mid h_2 = (h, a, o)).$$

Here q_0 denotes the posterior obtained when the independently sampled history is $h_2 = h$, whereas q_1 denotes the posterior obtained when $h_2 = (h, a, o)$.

Let

$$L(\theta) \triangleq p(o \mid h, a, \theta).$$

Since h_2 is independently sampled from the same world θ under policy π,

$$p(h_2 = (h, a, o) \mid \theta, \pi) = p(h \mid \theta, \pi)\, \pi(a \mid h)\, L(\theta),$$

Because a is fixed and $\pi(a \mid h)$ does not depend on θ, it gets canceled upon normalization. Therefore q_1 is obtained from q_0 by a likelihood tilt:

$$q_1(\theta) = \frac{q_0(\theta) L(\theta)}{\mathbb{E}_{\theta \sim q_0}[L(\theta)]} \tag{9}$$

$$= \frac{q_0(\theta) L(\theta)}{q(o \mid h)}, \tag{10}$$

where

$$q(o \mid h_2 = h) \triangleq \mathbb{E}_{\theta \sim q_0}\left[p(o \mid h, a, \theta)\right].$$

The double-take predictive can consequently be written as

$$q(o \mid h_2 = (h, a, o)) = \mathbb{E}_{\theta \sim q_1}[L(\theta)] \tag{11}$$

$$= \frac{\mathbb{E}_{\theta \sim q_0}[L(\theta)^2]}{\mathbb{E}_{\theta \sim q_0}[L(\theta)]}. \tag{12}$$

Hence

$$\mathrm{DT}(o; h, a) = -\log q(o \mid h_2 = h) + \log q(o \mid h_2 = (h, a, o)) \tag{13}$$

$$= \log \frac{\mathbb{E}_{\theta \sim q_0}[L(\theta)^2]}{\mathbb{E}_{\theta \sim q_0}[L(\theta)]^2}. \tag{14}$$

On the other hand, the Rényi divergence of order 2 between q_1 and q_0 is

$$D_2(q_1 \| q_0) = \log \int \frac{q_1(\theta)^2}{q_0(\theta)}\, d\theta \tag{15}$$

$$= \log \int q_0(\theta) \left(\frac{L(\theta)}{\mathbb{E}_{q_0}[L]}\right)^2 d\theta \tag{16}$$

$$= \log \frac{\mathbb{E}_{q_0}[L(\theta)^2]}{\mathbb{E}_{q_0}[L(\theta)]^2}. \tag{17}$$

Comparing with Eq. equation 14 gives

$$\boxed{\mathrm{DT}(o; h, a) = D_2(q(\theta \mid h_2 = (h, a, o)) \,\|\, q(\theta \mid h_2 = h))\,.}$$

Thus, the double-take reward is exactly the order-2 Rényi divergence between the two double-counted beliefs $q(\theta \mid h_2 = h)$ and $q(\theta \mid h_2 = (h, a, o))$.

B PROCEDURAL ENVIRONMENT GENERATION AND ORACLE WORLD INFORMATION SCORE

In DualLab, LilyPond, and BallPit, the maze within the procedural east room is generated by randomly connecting each cell in a sparse grid (odd-coordinate cells in the room's interior grid) to the cell north or west of it. Excluding the north and west edges of this 5×5 interior grid, each procedural layout amounts to 16 decisions, each of which can be observed when one of two possible connecting cells enters the agent's observation. These independent fair choices contribute one bit each when first revealed; fixed geometry and transient pond or monster dynamics contribute zero.

In LabTower, each ladder use generates a fresh maze with the same 16 independent choices. The sequence of mazes is shared across trials of a world, while monster motion is independent. We score each generation separately and sum the revealed bits over the episode: repeated views within one generation add nothing, while a new generation provides another 16 bits to discover.

In AlchemyLab, reset additionally samples a uniform permutation assigning six potion colors to six effects (± 2 on each of three vitals). Stepping onto a potion tile consumes it, and its visible effect identifies the assignment (provided the relevant vitals are not already saturated/depleted). After trying k distinct colors, $(6-k)!$ mappings remain, giving $\log_2(6!) - \log_2((6-k)!)$ potion bits. Adding the maze bits gives a maximum of $16 + \log_2(720) \approx 25.49$ bits; five colors suffice to identify the sixth by elimination.

To generate a CraftCamp layout, randomized Kruskal connects nine rooms, with a uniform corner entrance, a uniformly chosen farthest exit, and independent three-way terrain patches. We enumerate $1{,}048$ entrance/exit templates over 192 spanning trees, weighted by their generation probabilities, and 720 potion-effect permutations. The score is the decrease in posterior terrain-plus-potion entropy after the reset view, using first terrain views and compatible potion outcomes. Destructible stations are coarsened to floor; saturation ambiguity is retained and combat-contaminated health evidence omitted. This is exact for these evidence channels, not the full simulator.

Craftax-classic generates a 64×64 overworld using spatial noise and resource placement. We approximate its terrain distribution by Q, an equal mixture of horizontal and vertical center-rooted comb-tree Bayesian networks, fitted to $1{,}024$ independent generated maps and validated on 256 more. True positions register first observed tile values as evidence E_t; tree message passing marginalizes unobserved cells. We score $-\log_2 Q(E_t \mid E_0)$, conditioning on the reset view E_0. Repeated views and subsequent tile modifications add no evidence. This is an information proxy under Q, not exact native-generator information.

Craftax-full generates nine 48×48 floors using spatial-noise and room-and-corridor generators with distinct terrain and resources. We fit the same two-tree prior separately to each floor and sum their conditional terrain scores, ignoring cross-floor dependence. Only light-visible tiles enter the evidence, registered using true position and floor; hidden initial maps serve only to audit first observations. Both the reported score and the Craftax Oracle PPO reward cover terrain only. Potion mappings provide a separate $\log_2(720)$-bit information source that these curves and rewards omit.

Pretraining via Procedural Generation. Each family type combines a fixed geometry configuration with independent binary layout choices. The pretraining bank consists of 455 layouts for each of three families (branching trees, loopy mazes, and rooms and doorways) and three sizes (15×15, 21×21, and 27×27), totaling $4{,}095$ layouts. Branching-tree mazes are generated via randomized depth-first search, connecting each newly visited cell to the existing maze. Loopy mazes additionally remove selected walls to create alternative routes. Room mazes partition the arena into a 3×3 grid and randomly place doorways along boundaries. We sample distinct traversable start and goal cells, apply rotations and reflections, and reject any duplicate map layouts up to symmetry.

For exact information information measurement, evaluation uses separate generators and independent, fair binary choices controlling for connections, doorway positions, etc. Each choice affects a disjoint set of cells, whose values differ between its two alternative choices. Thus, observing any such cell identifies that choice completely, reducing its uncertainty from one bit to zero; unobserved choices remain independent and equiprobable.

Table 2: **Double Take reveals the most information in large heldout mazes.** Bits of world information revealed in out-of-distribution 201×201 sized mazes by 64,000 evaluation steps. Mean $\pm$ one standard deviation across four training seeds, with 80 trials per seed and family. Bold: best method.

Method Max. bits	Branching trees 1521	Winding corridors 49	45° diag. walls 3	Bottle-necks 2	Deceptive branches 33	Loopy mazes 760	Rooms & doorways 12	Diagonal walls 2	Dense diag. walls 4
E3B × RND	141.08 ± 38.47	1.26 ± 0.22	0.77 ± 0.34	1.24 ± 0.17	2.62 ± 1.45	407.30 ± 162.15	7.69 ± 1.91	1.28 ± 0.17	1.73 ± 0.36
RND	78.02 ± 60.28	0.55 ± 0.46	1.07 ± 0.72	0.65 ± 0.71	1.67 ± 0.71	119.52 ± 115.24	3.60 ± 3.28	0.72 ± 0.57	1.12 ± 0.94
ICM	97.45 ± 37.48	0.64 ± 0.31	1.21 ± 0.45	1.33 ± 0.19	1.50 ± 1.14	107.01 ± 163.58	5.23 ± 3.82	0.91 ± 0.67	1.20 ± 0.90
BYOL-Explore	76.90 ± 34.69	0.76 ± 0.45	1.43 ± 0.52	0.98 ± 0.37	0.48 ± 0.55	109.05 ± 68.47	6.70 ± 2.57	1.27 ± 0.15	1.48 ± 0.24
BYOL-Hindsight	82.55 ± 19.72	1.07 ± 0.22	1.28 ± 0.24	0.90 ± 0.16	1.38 ± 1.66	176.52 ± 86.62	5.33 ± 1.84	1.18 ± 0.17	1.30 ± 0.25
Double Take	$\mathbf{151.20 \pm 14.75}$	$\mathbf{1.84 \pm 1.42}$	$\mathbf{1.54 \pm 0.56}$	$\mathbf{1.74 \pm 0.34}$	$\mathbf{2.98 \pm 0.43}$	$\mathbf{445.33 \pm 47.85}$	$\mathbf{8.81 \pm 1.53}$	$\mathbf{1.52 \pm 0.21}$	$\mathbf{1.93 \pm 0.15}$

Let Θ denote the K binary layout variables, and let R_t denote those identified by observation evidence E_t (the collection of maze cell values the agent has seen up to time t, e.g. "cell (2,3) is a wall".) Then the remaining layout entropy is $K - |R_t|$ bits. Excluding information already visible at reset gives

$$I_t = H(\Theta \mid E_0) - H(\Theta \mid E_t) = |R_t \setminus R_0|$$

, with repeated observations of an identified choice containing no extra information.

The nine evaluated out-of-distribution families are:

- **Branching trees:** A tree with evenly spaced junctions, where each is connected to either its northern or western neighbor. Exploration reveals which connections are present.
- **Winding corridors:** Long horizontal corridors joined with alternating ends.
- **Bottlenecks:** Three thin open spots separated by two walls. Each wall has a passage either in its upper or lower end, requiring the agent to locate openings to proceed further.
- **Deceptive branches:** A direct start-to-goal corridor exists, with long alternating dead end branches.
- **Loopy mazes:** A connected lattice of corridors with additional passages in selected grid blocks. Each passage occupies one of two positions, creating alternating routes and repeated opportunities to detour the agent to revisit known regions.
- **Rooms and Doorways:** Nine rooms arranged in smaller square grids. Each grid contains a doorway at one of two possible locations, yielding twleve binary layout choices.
- **Diagonal walls:** Two thick, shallow-sloped walls divide an open area into three parts. Each barrier contains an opening at one of two possible horizontal positions.
- 45° **diagonal walls:** Three parallel 45° walls, each with an opening at one of two possible positions.

C Experimental Details

Observation. The agent never sees full maze. Instead, at each step, it observes a 7×9 window of the current grid centered at its current position, with each cell one-hot encoded by tile type (e.g. wall, floor) together with its (x, y) coordinates.

Policy. The actor and critic are separate networks, with no shared parameters. Each encode the current observation, which is a small CNN over 7×9 window concatenated with its coordinates, and feeds the result into its own 128-unit LSTM.

World Model Training. A GRU summarizes the history of past observations and actions, and from this summary, the WM predicts each cell of the next window as an independent categorical distribution over tile types. Each WM mini-batch trains with equal amounts of peer and self replay: Peer examples condition on an independent trajectory from the same world, while self examples

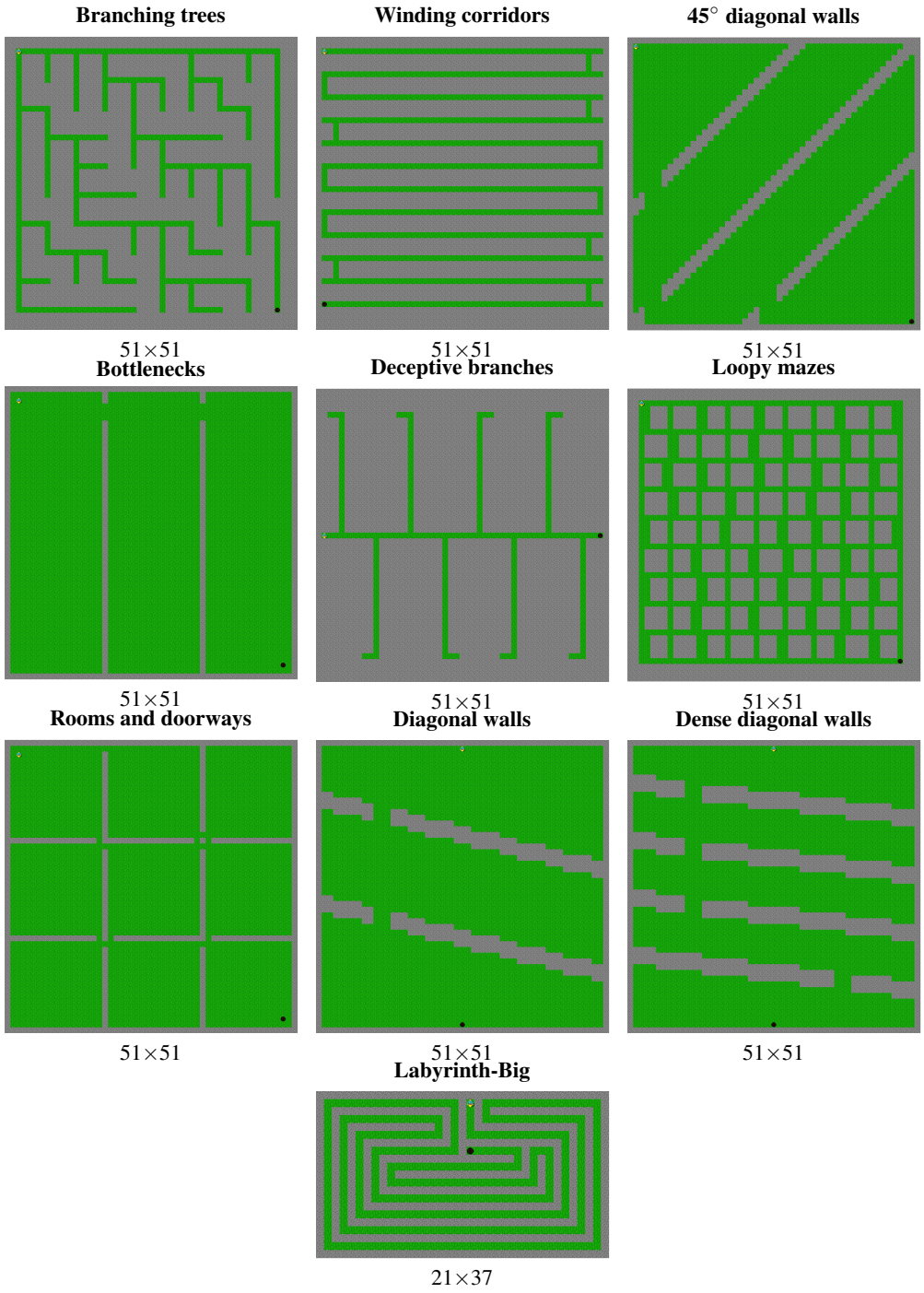


Figure 6: **Procedural maze families.** The nine held-out evaluation maze families and the fixed 21×37 Labyrinth-Big. Grass marks traversable cells, stone marks walls, the player marks the start and the ladder marks the goal.

condition on the agent's own earlier observations in current trajectory. The intrinsic reward is the reduction in the WM's negative log-likelihood of the observed next window once that observation is added to the evidence, clipped at $[0, 8]$ bits. See Table 4 for hyperparameter details.

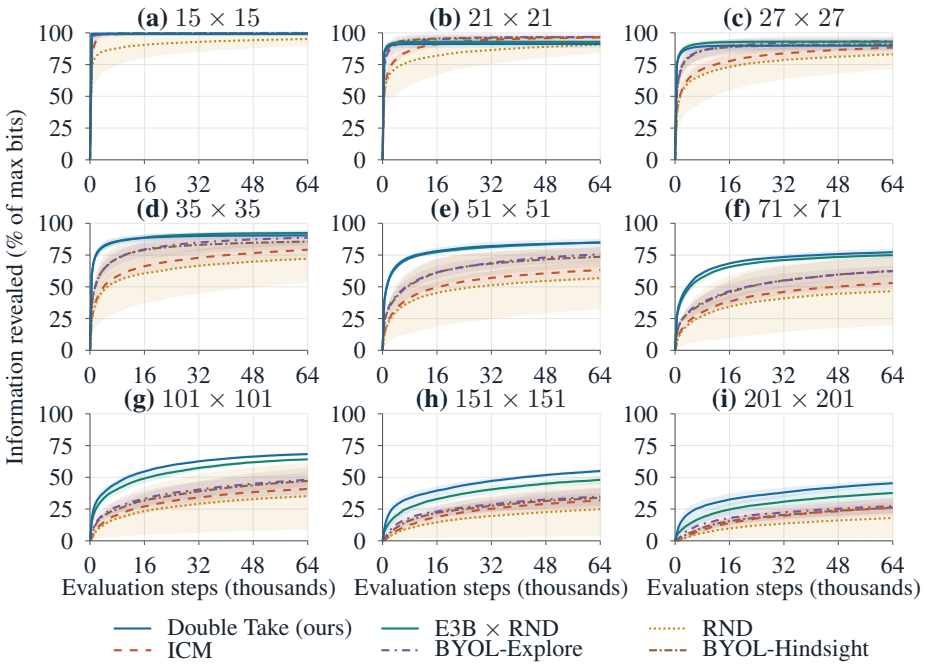


Figure 7: **Double Take's advantage grows with maze size.** Share of the available information revealed by frozen policies at each maze size, averaging seven families with equal weight (45° and dense diagonal walls are omitted because they are undefined at size 15). Each trial's information is divided by its maximum attainable bits before averaging. Lines show the mean over 4 training seeds and shading shows ±1 SD across seeds.

Table 3: Progress in the CraftCamp dungeon after 20.3M environment steps during evaluation in 2 trials of each of 64 held-out worlds × 5 training seeds, for 640 episodes total per method. The route from the start room to the exit room crosses three gates: a stone gate (mined with a wood pickaxe); an iron gate (stone pickaxe) into the forge room, whose chest holds one potion of each color; and a diamond gate (iron pickaxe) into the exit room, which two monsters guard. E3B × RND is the strongest baseline by world information. Mean ± one standard error over five training seeds; the last two rows count episodes. No method completes the dungeon.

	Double Take (ours)	Oracle PPO	E3B × RND
Mean per episode			
Rooms visited (of 9)	7.6 ± 0.1	6.7 ± 0.2	1.6 ± 0.1
Terrain information (bits)	17.6 ± 0.1	17.6 ± 0.2	5.1 ± 0.2
Potion information (bits)	0.4 ± 0.1	8.3 ± 0.5	0.0 ± 0.0
Share of episodes (%)			
Passed the stone and iron gates	96.6 ± 0.5	96.4 ± 1.6	0.0 ± 0.0
Reached the room before the diamond gate	90.8 ± 1.6	78.8 ± 4.7	0.0 ± 0.0
Visited all 8 rooms outside the exit room	77.7 ± 5.7	31.1 ± 4.8	0.0 ± 0.0
Made an iron pickaxe	11.1 ± 3.1	2.8 ± 1.7	0.0 ± 0.0
Opened the potion chest	6.7 ± 2.0	90.8 ± 5.0	0.0 ± 0.0
Episodes, of 640			
Opened the diamond gate	3	2	0
Entered the exit room	3	1	0

Online training pipeline: Training alternates between roll-out collection, PPO updates, and world model updates. The current policy collects trajectories, which are added to a finite first-in-first-out (FIFO) replay buffer, so newly trajectories replace the oldest ones when buffer gets full.

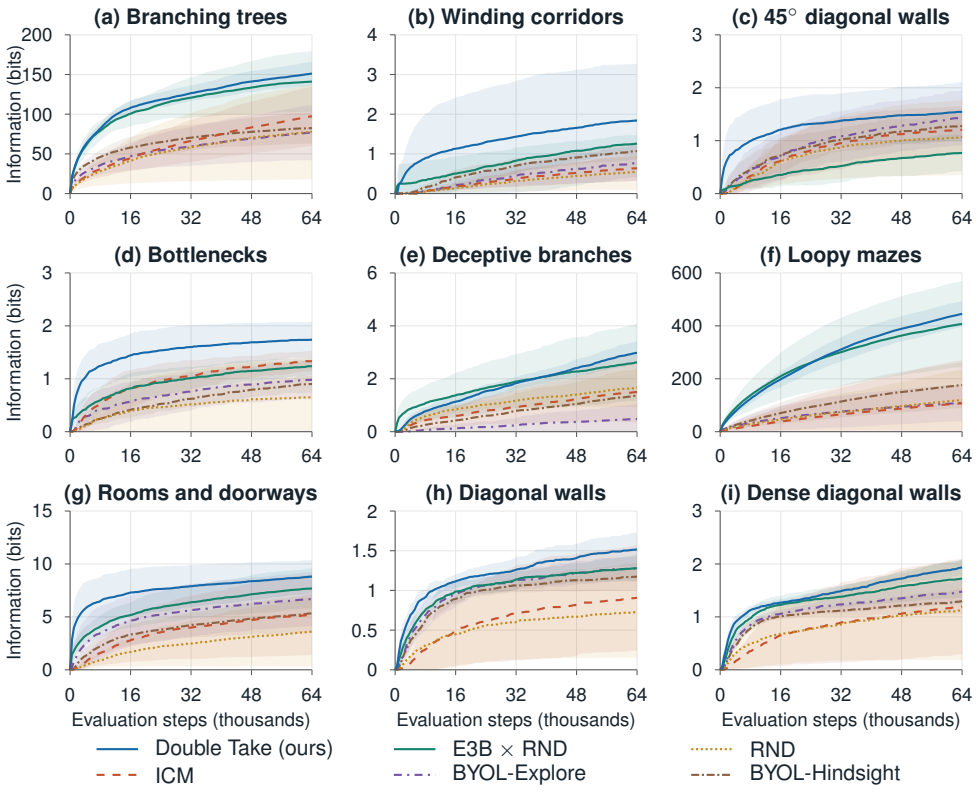


Figure 8: Information revealed over evaluation on 201×201 procedural mazes. Each panel is one held-out maze family; the y-axis is information revealed (bits) and the x-axis is evaluation steps (thousands). The lines show the mean over 4 training seeds (80 trials each), and shading shows ± 1 SD across seeds.

Table 4: Training hyperparameters for the Lab suite. Actor, critic, and world model use Adam; PPO settings are shared by all methods

PPO	
Learning rate	2×10^{-4}
Adam ϵ	10^{-5}
Discount γ	0.99
GAE λ	0.95
Clip ratio	0.2
Value clip	40
Entropy coef.	0.01
Value coef.	0.5
Max grad norm	0.5
Epochs	2
Minibatches	4
PPO BPTT length	1,984 (full episode)

World model	
Learning rate	3×10^{-4}
Max grad norm	1
Episodes per minibatch	16
Targets per episode	4
Updates per iteration	256
WM truncated-BPTT window	32
Replay buffer (pairs)	32
Collection	
Env. steps per iteration	15,872
Parallel envs (pairs)	8 (4)
Warm-up iterations	4

We only refine the policy with intrinsic reward for Double Take and all baseline methods, with PPO using only information gain as reward to compute advantages and update actor and critic. The world model is updated via incoming streams of mini-batches sampled from the replay buffer, using the peer and self prediction objectives described above.

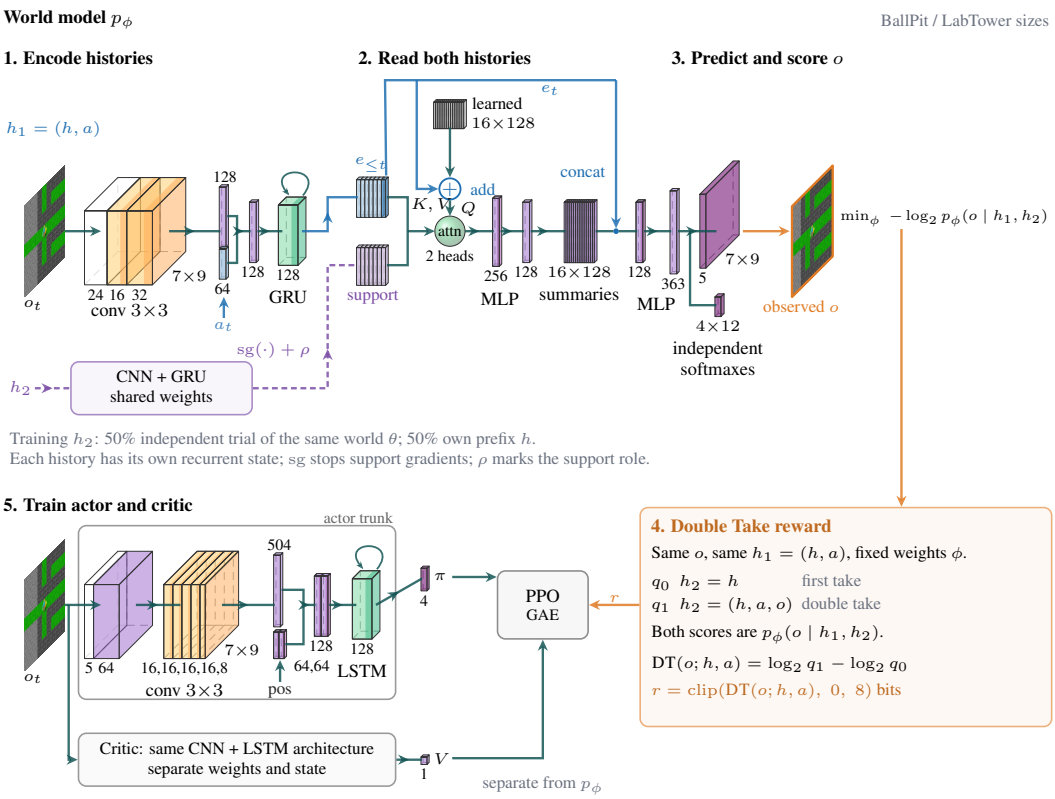


Figure 9: **Double Take world model and policy architectures.** In the world model (top), shared-world histories h_1 and h_2 are separately fed to a shared convolutional and recurrent encoder. The query latent e_t (an embedding of o_t from h_1) is added to 16 learned 128-dimensional query vectors to produce queries Q. These then attend to keys K and values L corresponding to the sequence of latents from the trajectories h_1 and h_2. h_2's history latents are added to a learned vector ρ to denote their acting as support, and a stop-gradient is applied to them. The policy and value networks are standard convolutional and recurrent architectures, with separate weights.

D AI USE STATEMENT

The paper's main body prose is fully human-written. We made extensive use of AI assistants (e.g. Claude Fable 5.1, Opus 5.5, ChatGPT 5.6 and 6) to write code, to organize and manage experiment sweeps across compute nodes, to verify our formal arguments, and to ideate on our algorithmic approach. We likewise used assistants to generate and iterate on figures, plots and tables (both using tikz and matplotlib), and to compile some of the implementation details in the appendix, followed by human review and editing.