REWARD-FREE EXPLORATION THROUGH CALIBRATED ORDERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning agents are often guided towards exploratory behavior with the help of intrinsic rewards. Maximizing the entropy of the state visitation distribution is a principled choice for such a reward. The objective pays the agent $-\log d$, where d is the visitation density. However, density estimation in high-dimensional spaces is unreliable. Furthermore, the shape of the latent space from which distances are measured is crucial, as it determines the semantic meaning of proximity, yet there is little methodology for selecting or validating such spaces, and we are left relying on assumptions about their semantic structure. We introduce CALOR (CALibrated ORdering), which extracts from the latent space only a pairwise *ordering* among states, thereby relying less on the exact geometry of that space. A predictor network is trained to reproduce that ordering with a pairwise ranking loss, and at the same time to place its own outputs on a *calibrated* scale, which restores the reward magnitude that an ordering alone leaves undetermined. The construction accepts any source of pairwise rarity comparisons, which we demonstrate by adding a second ordering derived from how long the agent has stayed alive. On reward-free Craftax-Classic at one billion interactions, CALOR reaches a Crafter score of 40.7 against 3.3 for the strongest baseline, and unlocks every one of the 22 achievements in every seed at least once. On three continuous-control exploration tasks it matches or exceeds every baseline, and on the humanoid maze it visits five times as many distinct states as the strongest of them. Ablations show that the ordinal signal and its calibration each contribute, and that neither on its own accounts for the result.

1 INTRODUCTION

Exploration is one of the central difficulties in reinforcement learning (RL) on long-horizon, sparse-reward tasks. A large body of work addresses it with *intrinsic motivation*, an internally generated reward that drives discovery even when task rewards are sparse or absent. A fully *reward-free* setting isolates this part of the problem by asking how much capable behavior an agent can acquire while optimizing an intrinsic signal alone, before the separate question of how to attribute and exploit an encountered reward. We study this question on Craftax-Classic (Matthews et al., 2024), the JAX reimplementation of the Crafter survival game (Hafner, 2022), at a budget of one billion interactions, following the reward-free protocol of C-TeC (Mohamed et al., 2026), which is a recently published reward-free result on this benchmark and our principal point of comparison. Craftax-Classic has a deep crafting *tech tree* in which most capabilities gate later ones, so random discovery scales poorly. An efficient method must therefore keep returning to the situations that are rarest in the agent's own experience, since that is where new capabilities can be found.

A natural objective for this setting is entropy maximization of the state-visitation density. If $d_\pi(s)$ is the visitation density at state s induced by the policy, then rewarding the agent with $-\log d_\pi(s)$ makes the expected return equal to the entropy of the states it visits, $H = -\int d_\pi(s) \log d_\pi(s)\, ds = \mathbb{E}_{s\sim d_\pi}[-\log d_\pi(s)]$. The challenge lies in obtaining $d_\pi(s)$.

Entropy maximization in the raw observation space is undermined by the fact that semantically similar states need not be close. For example, the same scene under day and night lighting may differ in most of its elements while differing hardly at all in what it means for the agent. A density intended to reflect rarity of *situations* should therefore be measured after an encoding step that carries some

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.

useful semantic meaning. A large body of exploration work rests on the assumption that a neural network trained to reconstruct or predict general aspects of the environment encodes observations into a latent space in which proximity carries semantic meaning that is useful for exploration (Liu & Abbeel, 2021; Seo et al., 2021). That assumption has driven progress in exploration research, but it is not guaranteed to hold in practice. A latent space is free to stretch one region and compress another, and its decoder can adapt at no cost, so the resulting density may not reflect the true rarity of states. Even an exact density measurement in such a space can therefore fail to be useful. Density estimation in high-dimensional latent spaces is also notoriously difficult because of the curse of dimensionality. Common proxies for density, such as nearest-neighbor distances, are correspondingly noisy and unreliable.

We present CALOR (CALibrated ORdering), which sidesteps the difficulties of high-dimensional density estimation by focusing only on *ordinal* comparisons of state rarity. Similar to prior work (Liu & Abbeel, 2021; Seo et al., 2021), CALOR obtains rarity comparisons from nearest-neighbor distances in a batch of on-policy observations encoded by an autoencoder, and trains a predictor network to reproduce the ordering. An ordering leaves every reward magnitude undetermined, so the predictor's outputs are also *calibrated* to a fixed target distribution over the agent's current experience. We select that target by a maximum-entropy argument on the reward values rather than on the visitation density. Among distributions on $\mathbb{R}_+$ with unit mean, the exponential $p(x) = e^{-x}$ has the largest entropy. In practice the predictor is calibrated against a Pareto distribution, which features a long tail for numerical stability and from which the exponential reward is recovered by a simple transformation. This fixes the reward marginal under every policy. Calibrating against the agent's current experience therefore also makes the rewards self-normalizing.

Curiosity-driven exploration seeks out situations that are surprising to the agent, but unlike a rarity-based objective it does not explicitly aim to return to states that hold no remaining surprise. We therefore argue for combining the two, since curiosity offers a focused signal towards what is currently surprising, while rarity encourages the agent to spread out across known states. CALOR consequently adds to the reward a curiosity-driven term based on the reconstruction error of the same autoencoder that supplies the latent codes.

We make three contributions. First, we introduce CALOR, a novel reinforcement learning exploration method that derives intrinsic rewards by calibrating a source of ordering among states. Second, we show that the ordering may be supplied by alternative comparators, by combining latent-distance and episode-age orderings under a single calibration. Third, we report large improvements in reward-free exploration on Craftax-Classic and on three continuous-control tasks, isolating the contribution of each component by ablation.

2 Related Work

The idea that an agent can generate its own learning signal from prediction error has long been central to formal accounts of artificial curiosity (Schmidhuber, 2010). ICM rewards error in a forward dynamics model defined over an inverse-dynamics feature space (Pathak et al., 2017), RND rewards error in predicting the output of a fixed random network (Burda et al., 2019), and C-TeC (Mohamed et al., 2026) extends the same idea from the next step to a distribution over future steps, which makes the bonus sensitive to long-horizon rather than single-step surprise.

Other approaches seek to maximize the entropy of a state-embedding distribution. For example, APT (Liu & Abbeel, 2021) and RE3 (Seo et al., 2021) do so through nonparametric estimators in which the log density at a sample is approximated from its distance to nearby samples. Count-based exploration instead rewards rarely visited states, extended to high-dimensional observations through pseudo-counts (Bellemare et al., 2016) and hashing (Tang et al., 2017), while NGU (Badia et al., 2020), NovelD (Zhang et al., 2021), BeBold (Zhang et al., 2020), E3B (Henaff et al., 2022), and DEIR (Wan et al., 2023) add episodic or elliptical bonuses. These share with entropy maximization the aim of rewarding rare states rather than surprising ones, but they are motivated as a proxy for uncertainty.

Skill discovery pursues diversity from a different direction, by maximizing mutual information between a sampled skill variable and the states the corresponding policy visits (Eysenbach et al., 2019). On Crafter itself the achievement hierarchy has also been targeted directly, through ob-

Algorithm 1 The CALibrated ORdering training loop with an arbitrary RL learner (e.g. PPO, SAC).

Initialize encoder $f_\phi : \mathbb{R}^{d_o} \to \mathbb{R}^{d_z}$, decoder $g_\psi : \mathbb{R}^{d_z} \to \mathbb{R}^{d_o}$, rarity predictor $h_\theta : \mathbb{R}^{d_z} \to \mathbb{R}$ and an RL learner with policy π ▷ observation dimension d_o and latent dimension d_z
while the interaction budget is not exhausted **do**
  $\mathcal{D}_{\mathrm{RL}} \leftarrow \mathrm{RLData}(\pi)$ ▷ $a_t \sim \pi(\cdot \mid \mathbf{o}_t), \quad \mathbf{o}_{t+1} \sim P(\cdot \mid \mathbf{o}_t, a_t)$
  for all $\mathbf{o}_i \in \mathcal{D}_{\mathrm{RL}}$ **do**
    $\mathbf{z}_i \leftarrow f_\phi(\mathbf{o}_i), \quad h_i \leftarrow h_\theta(\mathbf{z}_i), \quad P_i \leftarrow 1 + \log(1 + e^{h_i})$ ▷ Pareto support $P_i > 1$
    $r_i^{\mathrm{rar}} \leftarrow \alpha \log P_i, \quad r_i^{\mathrm{rec}} \leftarrow \|\mathbf{o}_i - g_\psi(\mathbf{z}_i)\|_2^2$
    $r_i^{\mathrm{int}} \leftarrow r_i^{\mathrm{rar}} + r_i^{\mathrm{rec}}$ ▷ intrinsic reward
  end for
  $\pi \leftarrow \mathrm{RLUpdate}(\pi, \mathcal{D}_{\mathrm{RL}}, r^{\mathrm{int}})$
  for all batches $\mathcal{B}$ of $\mathcal{D}_{\mathrm{RL}}$ **do**
    $n \leftarrow |\mathcal{B}|$
    $\mathbf{z}_i \leftarrow f_\phi(\mathbf{o}_i), \quad h_i \leftarrow h_\theta(\mathbf{z}_i), \quad P_i \leftarrow 1 + \log(1 + e^{h_i})$ for $i \in \mathcal{B}$
    $\delta_i \leftarrow \min_{j \in \mathcal{B},\, j \neq i} \|\mathbf{z}_i - \mathbf{z}_j\|_2$ ▷ rarity-ordering proxy
    $\sigma_{\mathrm{nn}} \leftarrow \mathrm{argsort}(\delta)$ ▷ densest first
    $L_{\mathrm{ord}} \leftarrow \frac{2}{n(n-1)} \sum_{i<j} \log\big(1 + e^{h_{\sigma_{\mathrm{nn}}(i)} - h_{\sigma_{\mathrm{nn}}(j)}}\big)$ ▷ RankNet rarity-ordering
    $(P_{[1]}, \ldots, P_{[n]}) \leftarrow \mathrm{sort}(P_1, \ldots, P_n), \quad u_k \leftarrow (k - \frac{1}{2})/n$
    $L_{\mathrm{cal}} \leftarrow \frac{1}{n} \sum_{k=1}^{n} \left(\log P_{[k]} + \frac{\log(1-u_k)}{\alpha}\right)^2$ ▷ Pareto-quantile
    update θ with $\lambda_{\mathrm{cal}} L_{\mathrm{cal}} + \lambda_{\mathrm{ord}} L_{\mathrm{ord}}$
    $L_{\mathrm{rec}} \leftarrow \frac{1}{n} \sum_{i \in \mathcal{B}} \|\mathbf{o}_i - g_\psi(\mathbf{z}_i)\|_2^2$
    update (ϕ, ψ) with L_{rec}
  end for
end while

jectives built around the achievements themselves (Moon et al., 2023) and through large language models prompted with a description of the game (Wu et al., 2023).

The ordering component of CALOR is the pairwise ranking objective introduced by RankNet (Burges et al., 2005). The calibration loss performs empirical quantile matching, the one-dimensional form of quadratic optimal transport (Villani, 2003; Peyré & Cuturi, 2019): sorted predicted log-rarities are matched to midpoint quantiles of the target exponential law.

3 Method: CALOR

3.1 Background

We consider a Markov decision process $(\mathcal{S}, \mathcal{A}, P, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(s' \mid s, a)$, and discount factor γ, but no reward function. The agent receives an observation o_t derived from the state s_t. A policy function $\pi(a_t \mid o_t)$ maps observations to a distribution over actions. The goal is to maximize the expected discounted return of an *intrinsic* reward generated by the exploration method, $\max_\pi \mathbb{E}_\pi[\sum_t \gamma^t r_t^{\mathrm{int}}]$. Extrinsic reward is used only during evaluation.

CALOR supplies an intrinsic reward to a standard RL learner. We use PPO (Schulman et al., 2017) in Craftax-Classic and SAC (Haarnoja et al., 2018) in the continuous-control experiments. Algorithm 1 abstracts the policy update and shows only the operations specific to CALOR.

3.2 Models and Losses

The encoder f_ϕ maps the flattened observation to a code $z = f_\phi(o) \in \mathbb{R}^{d_z}$, and the decoder g_ψ maps the code back to the observation. Both are residual multilayer perceptrons, and the pair is trained on reconstruction alone.

The predictor $h_\theta(z) \in \mathbb{R}$, also a residual multilayer perceptron, is trained to reproduce the ordering of observations by their rarity. This rarity ordering is determined by the distance to the nearest observation in the latent space. Given a permutation σ that lists a minibatch of n samples from

densest to sparsest, $h_\theta(z)$ is trained with the pairwise ranking loss of Burges et al. (2005), which smoothly penalizes every pair whose predicted rarity contradicts σ, without a margin.

The transformed predictor output $P_\theta(z) = 1 + \log(1 + e^{h_\theta(z)}) = 1 + \mathrm{softplus}(h_\theta(z))$ is calibrated to a Pareto rarity scale. This transformation places $P_\theta(z) > 1$ on the support of a Pareto distribution with unit scale. Within each batch, the calibration loss matches the sorted $P_\theta(z)$ values to fixed midpoint quantiles of that distribution.

3.3 Intrinsic Reward

The calibration targets a fixed reward distribution while retaining an unbounded upper tail for observations rarer than anything in the calibration batch. Under exact ordering and calibration, for observations o generated by policy π, $p = P_\theta(o)$ follows the unit-scale Pareto distribution with shape $\alpha > 0$. Let $u(p) = 1 - p^{-\alpha}$ be the unit Pareto CDF, so that $u(p) \sim \mathrm{Uniform}(0, 1)$. To obtain a rarity reward that follows a standard exponential distribution, as motivated in Section 1, we therefore define

$$r^{\mathrm{rar}}(o) = \alpha \log P_\theta(o) = \alpha \log(1 - u(p))^{-1/\alpha} = -\log(1 - u(p)) \sim \mathrm{Exp}(1).$$

The calibration consequently aims for a unit reward mean and variance. We use $\alpha = 2$, and ablate the effect of this choice in Section 7.

The intrinsic reward pairs the calibrated rarity with a curiosity term, $r^{\mathrm{int}} = r^{\mathrm{rar}} + r^{\mathrm{rec}}$, where r^{rec} is the reconstruction error of the same codec that supplies the codes. Its value is divided by a running root mean square of itself, which holds its scale near one as the raw error decays over training, and is then capped so that a single outlying observation cannot dominate the sum. Likewise, before computing nearest-neighbor distances or applying the rarity predictor, we subtract from each latent feature its running mean and divide by its running standard deviation, both estimated from recent codes. The decoder continues to receive the unstandardized code. This section deliberately omits some implementation details that we do not regard as part of the core idea but that matter in practice, and are therefore covered in Section 6. Appendix A provides the remaining implementation details.

4 Reward-Free Craftax-Classic

Craftax-Classic (Matthews et al., 2024) reimplements the Crafter survival game (Hafner, 2022) in JAX. We study it for the way its 22 achievements are arranged. Most of them gate later ones, wood before a wooden pickaxe, that pickaxe before stone, and so on. The probability of reaching a deep achievement at random is the product of the probabilities of every step before it. In the original benchmark an extrinsic reward marks each of those steps as it is unlocked, but in our reward-free setting simple random exploration fails to reach the deep achievements. To succeed, a method must substantially raise the marginal probability of each achievement along the chain. Every method trains for one billion interactions and never receives extrinsic reward, which is recorded for evaluation only.



Figure 1: **A rare Craftax-Classic state.** CALOR considers this a rare and rewarding state. The agent has added trees, furnaces, and stone to the map, it carries many tools, and a skeleton and cow are in view.

Baselines. Our principal point of comparison is C-TeC (Mohamed et al., 2026), whose recent reward-free Craftax protocol we follow. We also compare against ICM (Pathak et al., 2017) and RND (Burda et al., 2019) as standard prediction-error methods, and E3B (Henaff et al., 2022) as an episodic elliptical bonus designed for procedurally generated environments. Every method supplies its intrinsic reward to PPO and shares the environment, its wrappers, the interaction budget, and the rollout geometry with CALOR. Each baseline keeps its published configuration, which differs from ours in three settings. They use a recurrent policy, a discount of 0.99, and a linearly annealed learning rate, while CALOR uses a

feedforward policy, a discount of 0.999, and a constant learning rate. We add one control, labelled Alive, which replaces the intrinsic reward with a constant at every step the agent survives. Alive keeps everything else from CALOR's configuration, so it is primarily a control on our own agent rather than a comparison against the baselines.

In Figure 2 we report the Crafter score of Hafner (2022), the geometric mean of the per-achievement success rates aggregated over episodes, for which the same paper provides a human reference of 50.5. Every curve in this paper is a mean over five seeds with one standard deviation shaded. CALOR improves substantially on every baseline we ran, ending more than an order of magnitude above all of them. Every one of the 22 achievements is unlocked by every CALOR seed, and 17 of them are performed in at least half of all episodes at the end of training. Appendix B reports the final score of each method together with its episode return and per-achievement rates.

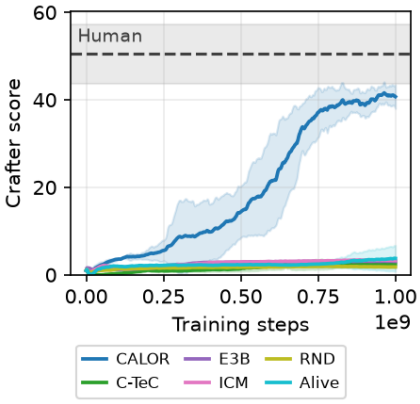


Figure 2: **Reward-free Crafter score comparison.** The human score of 50.5 is reported by Hafner (2022). Alive is the constant-reward control.

The Alive control. The Alive control, whose constant reward carries no information about which states are visited, reaches a Crafter score comparable with C-TeC and ICM. Like these baselines, it does not raise the probability of most achievements, focusing primarily on those related to survival, namely sleeping, eating, and drinking, and occasionally reaching other achievements that are shallow on the progression chain, as is evident from the lower Crafter score and per-achievement rates in the appendix. We propose this as a control because every method in the comparison has a strictly positive mean intrinsic reward over its whole run. This induces an incentive to stay alive, which can be counterproductive in environments where early termination is the goal. In such environments, normalization of the intrinsic reward to a zero or negative mean could be preferred. We argue that the choice of mean intrinsic reward, particularly in a reward-free setting, is an important design decision, and this constant can be considered as an extrinsic reward. These results show that the success of CALOR cannot be attributed solely to a positive mean intrinsic reward.

5 Continuous Control

In this section we investigate continuous-control environments, which introduce substantial changes to the setting. Observations and actions are continuous, the learner is SAC (Haarnoja et al., 2018) rather than PPO, and the exploration problem is physical reach rather than a crafting chain. These changes help establish whether CALOR can adapt to them. They also serve as a held-out test of the configuration, since CALOR is transferred with the defaults chosen on Craftax-Classic, and is tuned no further. The three tasks, Ant Hardest Maze, Humanoid U Maze, and Arm Binpick Hard, shown in Figure 3, are taken with their released configurations from C-TeC, reproducing their results. The baselines here are C-TeC, APT (Liu & Abbeel, 2021), ICM, and RND. APT replaces E3B in this comparison for two reasons. E3B trains its embedding with an inverse-dynamics classifier over a discrete action set, so it does not transfer to continuous actions without changing the method itself, and its bonus is episodic in a way that suits procedurally generated maps rather than a fixed arena. APT, on the other hand, is the nearest-neighbor state-entropy method closest to CALOR's rarity term, which makes it the most informative comparison we can run on these tasks. Unlike on Craftax-Classic, every method runs in that release's SAC scaffold with only the intrinsic module and the reward

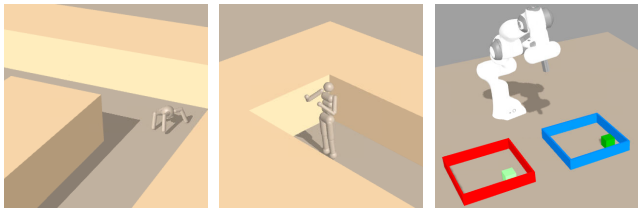


Figure 3: **Visualization of continuous-control environments.** The evaluated environments from left to right: Ant Hardest Maze, Humanoid U Maze, and Arm Binpick Hard.

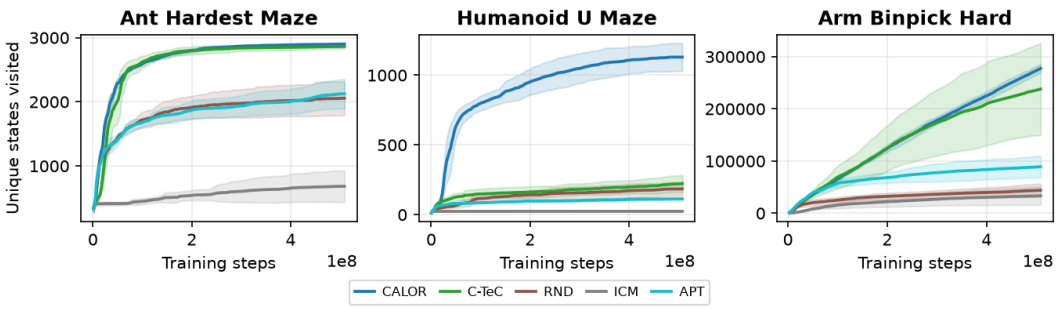


Figure 4: **Continuous-control exploration comparison.** Distinct states visited on the three continuous-control exploration tasks.

it produces replaced, so the discount, the learning rate, the feedforward actor and critic, and the rollout and replay geometry are identical across methods. Each run is half a billion interactions, task and dense rewards are disabled throughout, and the reported metric is the number of distinct states visited after discretizing torso or cube position with the same discretization code as in the original C-TeC release.

Figure 4 reports the number of distinct states visited, and Appendix B gives the final value reached by each method. On the ant maze CALOR and C-TeC reach similar performance, and the remaining three methods fall well short of both. On the humanoid maze CALOR visits five times as many distinct states as the strongest baseline (C-TeC). On the robot arm the two leading methods overlap, with CALOR reaching the higher mean while C-TeC's spread covers it. CALOR's spread across seeds is much narrower, and its curve is still close to straight at the budget limit where C-TeC's has begun to bend.

6 Additional Method Components

In this section we discuss three important components of CALOR that are not part of its core design but still affect its performance. Figure 5 reports the Crafter score when each one is removed individually.

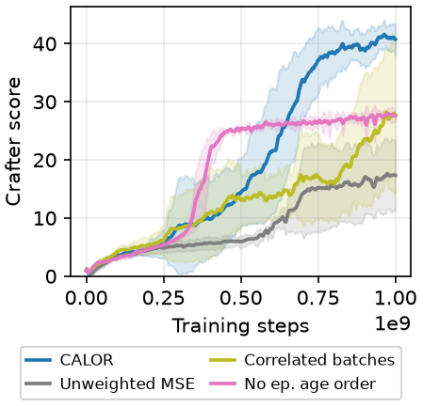


Figure 5: **Ablation of additional method components.** Each of the three components of Section 6 is removed from the default configuration.

Error-weighted reconstruction. The rarity ordering and the curiosity term are both read off the codec, so the exploration signal can be no better than what the codec captures. During evaluation we observed that cows, which are not a rare sight, were reconstructed visibly poorly. To address this, we introduce error-weighted reconstruction, weighting each per-element squared error by its own detached and mean-normalized value. This moves capacity toward whatever the codec currently reconstructs worst. Formally, let $\mathbf{\Delta}$ denote the vector of per-element squared reconstruction errors, with $\Delta_k = \|[\mathbf{o} - g_\psi(f_\phi(\mathbf{o}))]_k\|_2^2$.

$$L_{\text{rec}} = \frac{1}{m_o} \sum_{k=1}^{m_o} w_k \, \Delta_k, \qquad \mathbf{w} = \frac{\text{sg}[\mathbf{\Delta}]}{\text{mean}\left(\text{sg}[\mathbf{\Delta}]\right)},$$

where m_o is the number of observation elements and sg (stop gradient) marks a quantity held constant during backpropagation. The weights average to one, so they redistribute the gradient across observation elements without rescaling it. This change allowed the codec to better reconstruct rarer observations, which in turn improved the overall exploration signal. We do not consider this part of CALOR, but we encourage practitioners applying CALOR to ensure that the codec adequately reconstructs all observation elements, especially the rarer ones. A stronger codec, for instance one

whose convolutional structure matches the layout of the observation, may reconstruct the rare elements well enough that this correction becomes unnecessary. Similar weighting was *not* applied to the reconstruction reward, since its only purpose is to help train the codec.

Decorrelated minibatches. The nearest-neighbor estimate should ideally be computed with respect to independently sampled observations from the current visitation distribution. In practice, it is computed from the current minibatch, making it important to limit correlations among its observations. Because nearby observations from the same environment trajectory are often strongly correlated, we construct each CALOR minibatch by drawing exactly one observation from every parallel environment. Before forming the minibatches, we independently permute the timesteps within each environment's rollout. Thus, every rollout observation is used exactly once, while each minibatch contains at most one observation from any environment trajectory. In our Craftax-Classic experiments, the number of parallel environments equals the CALOR minibatch size, making this construction straightforward. Other sampling schemes could provide the same decorrelation property when these quantities differ. Although recommended in practice, this decorrelation strategy was not employed in the continuous-control tasks, where consecutive observations are less strongly correlated because of the nature of the environment dynamics.

Episode-age ordering. To illustrate that the ordering can be supplied from different sources, on Craftax-Classic we add a second ordering loss to the same predictor, in which observations are ranked by how many steps their episode has lasted rather than by latent distance. Both orderings train the predictor jointly and share its single calibration, so the predictor parameters θ are updated with $\lambda_{\text{cal}} L_{\text{cal}} + \lambda_{\text{ord}} L_{\text{ord}} + \lambda_{\text{age_ord}} L_{\text{age_ord}}$. The motivation is that a longer-lived episode is more likely to contain rare events. Craftax-Classic in particular contains one-way transitions such as crafting a tool, which cannot be undone once made, so their consequences can only appear later in an episode. It should nonetheless be added with care. Section 7 shows that keeping this ordering while removing the reconstruction reward collapses the agent.

7 CALOR Component Ablations

In this section we examine the importance of the individual components of CALOR in order to validate the inclusion of each one.

The importance of calibration and ordering. Without any ordering loss the predictor is still calibrated but no longer tied to nearest-neighbor structure, so it emits a correctly distributed but uninformative rarity and the score falls, as we can see in Figure 6. Without the calibration loss the ranking survives but its scale is unconstrained, and the score also falls. Both these ablations have a large impact, which highlights the importance of the two central components of CALOR.

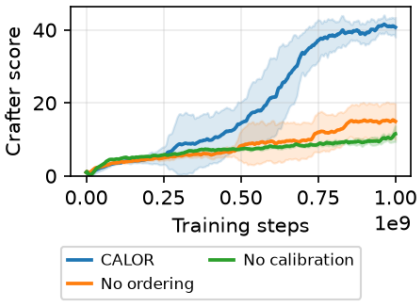


Figure 6: **Calibration and ordering ablations.**

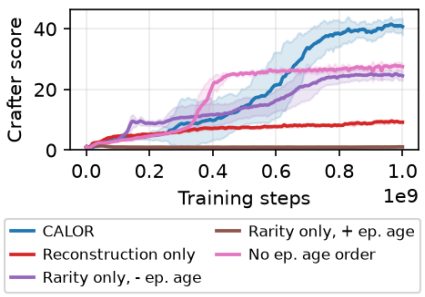


Figure 7: **Reconstruction and episode-age ablations.**

The importance of reconstruction reward. In Figure 7 we see that "Reconstruction only" reaches a relatively low score, so the calibrated ordinal signal is the larger contributor by a wide margin. Its score of 9.2 is also well above the constant-reward floor of 3.8 from Section 4, so it is an exploration signal in its own right. Removing this reward also leads to two further observations. First, in "Rarity only, + ep. age" we remove only the reconstruction reward while keeping the episode-age ordering loss. This is highly detrimental, and the method collapses. We believe that episode age introduces a stiffness whereby otherwise common states are considered rare, because the policy does not diversify and late-episode states still contain these common ones. The reconstruction reward gives the agent an initial signal

towards novel states, which helps it escape this negative cycle. Second, in "Rarity only, - ep. age" we additionally remove the episode-age ordering loss, and in "No ep. age order" we remove only the episode-age ordering, as also seen in Figure 5. In this scenario, the reconstruction reward makes a small relative contribution. We believe the explanation for either behavior to be environment-specific, so we conclude that these two components, the episode-age loss and the reconstruction reward, deserve consideration when applying CALOR, much like the choice of values for λ_{cal}, λ_{ord} and $\lambda_{\text{age_ord}}$.

Pareto shape. Figure 8 shows very poor performance with shape 1, but little difference between shapes 2 and 3. Regardless of the Pareto shape, the reward remains exponentially distributed with unit mean, so the shape is largely a matter of the predictor's learning dynamics. A unit-shape Pareto distribution has infinite variance and a particularly heavy tail, which may cause the predictor to take longer to settle. We use shape 2 throughout because it retains a substantial tail, leaving room to distinguish several rare states as problem size grows, without the more extreme conditioning of shape 1.

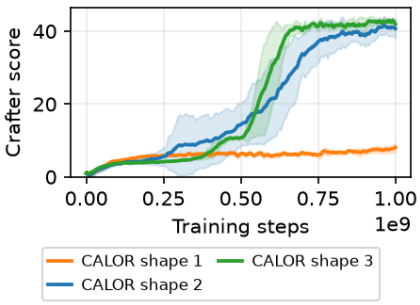


Figure 8: **Pareto shape ablation.**

8 Concluding Remarks

8.1 Analysis of Results

CALOR provides substantial performance improvements in reward-free exploration. What stands out is how it inflates the marginal probabilities of almost all achievements, as measured by the Crafter score, by rewarding them over time. We have not observed similar emergent behavior from other methods. The large performance gap in Humanoid U Maze likely has the same origin as the one on Craftax-Classic. CALOR does not only discover but likely also promotes escaping the maze to explore outside of it, whereas C-TeC discovers but does not encourage this behavior, much as CALOR promotes the `collect_stone` achievement in Craftax-Classic so that its agent reaches it in 98.50% of episodes, while C-TeC reaches it in only 1.63% of episodes. However, we did not verify this. The Ant Hardest Maze environment is likely saturated for both C-TeC and CALOR, and a different benchmark should be considered moving forward. It appears Arm Binpick Hard would benefit from more training time, since CALOR's performance shows little sign of approaching a plateau. It would be very interesting to run this environment for longer, although it is the slowest of all our tested environments. Finally, on Craftax-Classic CALOR seems likely to approach a plateau, with limited further improvements in the rarest achievements despite continued training. Likely causes are the limitations of the agent's memory and representational capacity. The environment also offers several distractors that compete for the agent's attention and can hinder the promotion of the rarest achievements, which become no more interesting than one of the many constructions the agent can build on the map.

8.2 Limitations

Comparison of methods in reinforcement learning is often difficult because inherent variance and seemingly minor implementation choices can materially change the results (Henderson et al., 2018; Engstrom et al., 2020). Our comparisons contain several potential confounds. Some aspects of CALOR are not shared by the baselines, such as the components from Section 6 and the differences in hyperparameters from Section 4. We believe some of these to be potential improvements for any method, like the decorrelation of batches for any neighbor-comparing method, some to be temporary solutions, like the weighted reconstruction loss, and some to be changes that would make only minor differences, like the learning-rate schedule. We also have some differences between the CALOR implementations for Crafter and for the continuous-control environments, as discussed in Section 5. In Crafter, there are more differences to the baselines than in the continuous-control environments. Therefore, part of the gap in Figure 2 may belong to the agent rather than to the signal. We did not try to improve on previous methods, as this may introduce further confounds. Since they were run on the same benchmark, we left their parameters untouched. Even though these differences make

for a less unified comparison, they have the benefit of testing CALOR in more varied situations. Our main contribution is the calibrated ordering concept which we encourage practitioners to experiment with in different ways.

Another limitation, shared with other nearest-neighbor estimators, is that the batches from which the estimates are formed should be reasonably large to reduce variance, while computation is quadratic in the ranking batch size. In practice, with our batch size of 1024, running once per minibatch, the runtime impact is small beside the PPO update.

8.3 Future Work

We propose several directions for future work:

1. **Alternative sources of ordering** For example, replacing the ordering by distance in the latent space with ordering by temporal distance. Our episode age ordering is a first step towards this idea. An estimation of distance in time between states could prevent the collapse observed in Figure 7, since it would discourage the agent from remaining in the same place. Observations of high episode age would no longer be considered rare unless they were also distant in time from common ones.
2. **Advanced curiosity driven rewards** Our current reconstruction reward is a simple form of curiosity-driven reward. Future work could explore more sophisticated curiosity mechanisms, such as prediction error in a learned dynamics model. These models would likely enrich the semantic capabilities of the agent, which could be beneficial also for the ordering loss, but in Crafter they would also introduce noise because of its partial observability. Exploring the map is a source of unpredictable noise which is known to challenge curiosity-driven approaches. We chose to sidestep this problem to reduce confounds in this work by removing the noise altogether. Integration with curiosity driven methods designed to filter out noise, such as Blad et al. (2026) or Hou et al. (2026), are natural extensions.
3. **Weighted ordering loss** The presented ordering loss weights all $n(n-1)/2$ pairs equally, which spends most of its gradient separating common states from each other, even though the top of the ranking is the most relevant part for an exploratory reward signal. Weighting pairs by their effect on the ranking as LambdaRank does for retrieval metrics (Burges et al., 2006) is a promising variant we have not run.
4. **Alternative reward distribution** Our target reward distribution is chosen as the maximum-entropy law over positive rewards with a fixed mean. Given real-valued rewards, fixed mean, and fixed variance, the maximum entropy distribution would instead be Gaussian. Calibrating to a Gaussian is therefore a principled alternative to our current choice of reward distribution. Similarly, calibrating directly to the exponential distribution instead of using the Pareto parameterization could be another approach worth investigating.

AI Use Statement

Generative AI tools were used for language editing, LaTeX formatting and coding assistance. The authors reviewed and verified all AI-assisted changes and take responsibility for the final content of this work.

Ethics Statement

This work involves no human subjects, no personal data, and no data collection of any kind, since all experiments are run in simulated environments that ship with their own generated content. Nevertheless, exploration methods are dual-use. When deployed at scale in real or high-fidelity simulated environments, an agent seeking unfamiliar states may discover and exploit unsafe behaviors, causing material harm even without malicious intent. As increasingly capable world models make it possible to search larger spaces, they may amplify both deliberate misuse and unintended consequences. We therefore encourage careful and responsible deployment wherever exploratory agents could affect people or real-world systems.

Reproducibility Statement

Appendix A documents the full experimental setup. The accompanying source code provides the experiment launchers and the scripts used to produce the figures. All results use five seeds, 0 through 4. Code for reproducing all experiments is available at `https://github.com/[to-be-filled-upon-publication]`.

References

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Sye57xStvB.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1471–1479, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Samuel Blad, Martin Längkvist, and Amy Loutfi. Measuring learning progress via gradient-momentum coupling. *CoRR*, abs/2605.05856, 2026. doi: 10.48550/ARXIV.2605.05856. URL https://doi.org/10.48550/arXiv.2605.05856.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In Luc De Raedt and Stefan Wrobel (eds.), *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pp. 89–96. ACM, 2005. doi: 10.1145/1102351.1102363. URL https://doi.org/10.1145/1102351.1102363.

Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 193–200. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/af44c4c56f385c43f2529f9b1b018f6a-Abstract.html.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. *CoRR*, abs/2005.12729, 2020. URL https://arxiv.org/abs/2005.12729.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018. URL http://arxiv.org/abs/1812.05905.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=1W0z96MFEoH.

Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via elliptical episodic bonuses. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/f4f79698d48bdc1a6dec20583724182b-Abstract-Conference.html.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3207–3214. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11694. URL https://doi.org/10.1609/aaai.v32i1.11694.

Zhibo Hou, Zhiyu An, and Wan Du. Beyond noisy-TVs: Noise-robust exploration via learning progress monitoring. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wzm38DRLhC.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 18459–18473, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/99bf3d153d4bf67d640051a1af322505-Abstract.html.

Michael T. Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Thomas Jackson, Samuel Coward, and Jakob Nicolaus Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35104–35137. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/matthews24a.html.

Faisal Mohamed, Catherine Ji, Benjamin Eysenbach, and Glen Berseth. Temporal representations for exploration: Learning complex exploratory behavior without extrinsic rewards. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KjYpHySlb0.

Seungyong Moon, Junyoung Yeom, Bumsoo Park, and Hyun Oh Song. Discovering hierarchical achievements in reinforcement learning via contrastive learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/c919a2b5ec1de69f2629f9119676e336-Abstract-Conference.html.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017. URL http://proceedings.mlr.press/v70/pathak17a.html.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019. doi: 10.1561/2200000073. URL https://doi.org/10.1561/2200000073.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Trans. Auton. Ment. Dev.*, 2(3):230–247, 2010. doi: 10.1109/TAMD.2010.2056368. URL https://doi.org/10.1109/TAMD.2010.2056368.

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

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9443–9454. PMLR, 2021. URL http://proceedings.mlr.press/v139/seo21a.html.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 2753–2762, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3a20f62a0af1aa152670bab3c602feed-Abstract.html.

Cédric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, 2003. ISBN 978-0-8218-3312-4. doi: 10.1090/GSM/058. URL https://bookstore.ams.org/gsm-58.

Shanchuan Wan, Yujin Tang, Yingtao Tian, and Tomoyuki Kaneko. DEIR: efficient and robust exploration through discriminative-model-based episodic intrinsic rewards. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 4289–4298. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/477. URL https://doi.org/10.24963/ijcai.2023/477.

Yue Wu, So Yeon Min, Shrimai Prabhumoye, Yonatan Bisk, Russ Salakhutdinov, Amos Azaria, Tom M. Mitchell, and Yuanzhi Li. SPRING: studying papers and reasoning to play games. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/46c2a9a6f2b2be68682013eb1173c801-Abstract-Conference.html.

Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E. Gonzalez, and Yuandong Tian. Bebold: Exploration beyond the boundary of explored regions. *CoRR*, abs/2012.08621, 2020. URL https://arxiv.org/abs/2012.08621.

Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E. Gonzalez, and Yuandong Tian. Noveld: A simple yet effective exploration criterion. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 25217–25230, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d428d070622e0f4363fceae11f4a3576-Abstract.html.

A Implementation Details

A.1 Craftax-Classic

Hyperparameters. Table 1 lists the default CALOR configuration used for the main result. Its settings come from manual runs on this environment during CALOR's development, while attempting

Group	Hyperparameter	Value
General	Numerical floor ϵ	1×10^{-8}
PPO	Parallel environments N	1024
PPO	Rollout length T	64
PPO	Rollout batch $B = N\,T$	65536
PPO	Minibatches M	8
PPO	Minibatch size B/M	8192
PPO	Update epochs E_π	4
PPO	Discount γ	0.999
PPO	GAE λ	0.8
PPO	Clip ϵ_{clip}	0.2
PPO	Value loss weight λ_V	0.5
PPO	Entropy bonus weight λ_{ent}	0.01
PPO	Learning rate	2×10^{-4}
PPO	Learning-rate anneal	off
PPO	Max gradient norm	1.0
PPO	Activation	leaky ReLU
PPO	Hidden size	512
PPO	Optimistic reset ratio	16
PPO	Return-norm EWMA weight	0.999
Codec	Learning rate	1×10^{-4}
Codec	Adam β_1	0.99
Codec	Hidden size	512
Codec	Layers	5
Codec	Latent size m	512
Codec	Latent activation	identity
Codec	Exploration update epochs E_{enc}	1
Codec	Codec passes per epoch E_{codec}	2
Codec	Reconstruction loss	error-weighted MSE
Predictor	Learning rate	2×10^{-4}
Predictor	Hidden size	512
Predictor	Layers	5
Predictor	Ranking batch size n	1024
Predictor	Code standardization EWMA decay	0.999
Predictor	Pareto shape α	2.0
Predictor	Calibration loss weight λ_{cal}	0.01
Predictor	Rarity-ordering loss weight λ_{ord}	1.0
Predictor	Episode-age-ordering loss weight $\lambda_{\text{age_ord}}$	10.0
Reward	Reconstruction reward cap $c_{\max}$	10.0
Reward	Reconstruction EWMA decay β	0.999

Table 1: **Craftax-Classic hyperparameters.** Default CALOR hyperparameters on Craftax-Classic.

to minimize changes for comparability with the baselines, and no automated or exhaustive hyperparameter search was conducted. Every Craftax-Classic run uses five seeds, 0 through 4, and one billion environment interactions, which is 15,258 PPO updates. One such run takes about 5.2 hours at 53.8k interactions per second on a single RTX 3080.

Baselines. The four baselines are local ports rather than byte-for-byte executions of the upstream scripts, because this repository supplies its own environment and logging adapters. Each port is pinned and audited against the C-TeC authors' public release, and its algorithmic and training-loop details are resolved against that code with the published configuration tables taking precedence where the release is internally inconsistent. C-TeC uses a recurrent policy with a 512-unit GRU, three 1024-unit encoder layers, a normalized 64-dimensional representation with a learned temperature, an L2 InfoNCE objective, a temporal discount of 0.3, and the discounted Monte-Carlo temporal-distance reward. E3B uses a recurrent policy with a 32-dimensional inverse-dynamics embedding over three 256-unit layers and an episodic elliptical bonus. ICM uses a 512-dimensional embedding over two 256-unit layers with forward and inverse losses, and RND a 512-dimensional

output over two 256-unit layers. All four run for the same one billion interactions with the same wrappers and rollout geometry as CALOR, and none receives task reward.

Architectures. The encoder, decoder, and predictor are residual multilayer perceptrons. Each network projects its input to the hidden width once, then applies pairs of layers as pre-normalized residual branches, and finishes with a linear output projection.

Nearest-neighbor computation. Nearest-neighbor distances are computed exactly over the full minibatch of 1024 codes, in chunks of 128 query rows so that the pairwise distance matrix is never materialized in full, for runtime optimization purposes. The pairwise ordering loss is evaluated the same way.

Scale of the curiosity term. The rarity term holds its scale by construction, but the raw reconstruction error falls as the codec improves. A fixed mixture would therefore change meaning over training. We normalize and cap the reconstruction reward before adding it,

$$r^{\mathrm{rec}}(o) = \min\left(\frac{e(o)}{\sqrt{\bar{m}}},\, c_{\max}\right), \qquad e(o) = \frac{1}{m_o}\|o - g_\psi(f_\phi(o))\|_2^2,$$

where $\bar{m}$ is an exponential moving average of e^2 across rollouts with decay 0.999 and $c_{\max} = 10$. The normalization keeps the root mean square near one, allowing the unweighted sum of reward components in Section 3, while the cap prevents one outlying observation from dominating it.

Encoding normalization. The decoder operates on the raw code z, while nearest-neighbor distances and the predictor use

$$\widetilde{z}_k = \frac{z_k - \mu_k}{\sqrt{v_k + 10^{-6}}}.$$

On Craftax-Classic, μ_k and v_k are debiased exponential moving estimates of each code feature's mean and variance with decay 0.999. They are refreshed from the full rollout after the encoder update and held fixed during the next rollout. The continuous-control implementation instead computes the feature statistics within the current training batch.

PPO objective. The actor and critic are separate three-hidden-layer MLPs, each of width 512, trained jointly with a single optimizer and learning rate. The critic regresses online-normalized value targets with a clipped value loss, advantages are normalized within each minibatch, and the optimized objective is the clipped policy-gradient loss plus a value loss weighted by λ_V and an entropy bonus weighted by λ_{ent}.

Training loop. Each update collects a rollout of $B = NT$ transitions from N parallel environments run for T steps, computing the intrinsic reward online as it goes. After the rollout, the actor-critic is updated for E_π epochs over M minibatches, and the codec and predictor are updated for E_{enc} epochs over the cross-environment minibatches of Section 6. Within a codec epoch, the first $E_{\mathrm{codec}} - 1$ passes update the encoder and decoder on reconstruction alone, and the final pass updates the encoder, decoder, and predictor together, with the predictor operating on detached codes. The per-feature standardization statistics are refreshed from the updated encoder at the end of each update, so the reward computed during the next rollout uses statistics that are one update old.

Interaction-count comparability. Craftax-Classic is designed for highly parallel execution, and we follow its reference configuration of 1024 parallel environments. We observed that this degree of parallelism lowers sample efficiency, so interaction counts in our experiments are not directly comparable to those reported for the original Crafter implementation.

A.2 Continuous Control

Every learned method uses the authors' released SAC training, replay, and evaluation scaffold and the hyperparameters of their tables, namely a 1000-step horizon, 62-step unrolls, two 256-unit feed-forward actor and critic layers, a discount of 0.99, and learning rates of 3×10^{-4}. These are identical for every method, so unlike on Craftax-Classic nothing outside the intrinsic module differs between

them. Ant Hardest Maze and Arm Binpick Hard use 1024 parallel environments with batch size 1024, and Humanoid U Maze uses the release's 256 and 256. The five-hundred-million-interaction budget is the published one for Ant Hardest Maze and Humanoid U Maze, and twice the published one for Arm Binpick Hard.

The policy receives the physical state but never the sampled task goal, so the intrinsic model may consume the complete physical state without leaking a target to the policy. The reported metric is unique discretized coverage at the release's bin width of 0.5, over Ant Hardest Maze torso (x, y), Humanoid U Maze torso (x, y, z), and cube (x, y, z) for Arm Binpick Hard, accumulated from replay samples at each evaluation exactly as the released learners do.

CALOR is inserted into the same scaffold by replacing only the intrinsic module and the reward it produces. Its encoder, decoder, and predictor use a 32-dimensional latent and four 256-unit hidden layers, and everything else follows Section 3. Unlike on Craftax-Classic, where the discrete symbolic observations are reconstructed in their native encoding, here each continuous observation feature is standardized by a running mean and variance before it is passed to the encoder and used as the reconstruction target. This prevents physical quantities with larger numerical scales from dominating the reconstruction loss. The episode-age ordering is disabled throughout. Two upstream launcher discrepancies are corrected in the pinned runtime, in that published model sizes supersede inconsistent shell defaults, and APT uses the paper's Gaussian augmentation standard deviation of 0.5 rather than the 0.3 hard-coded in the released loss file.

Taking the median throughput of each CALOR run and averaging over seeds gives 36.4k, 16.4k, and 4.4k interactions per second on Ant Hardest Maze, Humanoid U Maze, and Arm Binpick Hard, so a five-hundred-million-interaction run takes about 3.8, 8.5, and 31 hours respectively on an NVIDIA GH200 node.

B Detailed results

Table 2 gives the final Crafter score and episode return of each method on reward-free Craftax-Classic, and Table 3 gives the success rate of each of the 22 achievements at the end of training, as a percentage of episodes, averaged over the final 100 PPO updates and then over five seeds. Each update first collects a rollout of $1024 \times 64 = 65{,}536$ interactions, and its logged achievement rate is the percentage among episodes completed during that rollout. Consequently the window covers $6{,}553{,}600$ interactions per seed.

Method	Crafter score	Episode return
CALOR	40.7 ± 2.7	14.8 ± 0.3
ICM	3.3 ± 0.3	2.8 ± 0.2
E3B	3.1 ± 0.2	2.9 ± 0.1
C-TeC	2.5 ± 0.9	3.6 ± 0.9
RND	1.8 ± 0.2	0.8 ± 0.2
Alive	3.8 ± 2.9	3.8 ± 1.8

Table 2: **Final Crafter score and episode return.** Final Crafter score and episode return on reward-free Craftax-Classic at one billion interactions. Alive is the constant-reward control of Section 4, which shares CALOR's configuration.

Every one of the 22 is unlocked by every CALOR seed at some point during training. Diamond in particular is listed with a rate of 0.00 because it collected 27, 5, 17, 19, and 1 times by the five seeds over roughly 1.5 million episodes each, which rounds to zero as a success rate.

Score definitions. We report the Crafter score of Hafner (2022), $S = \exp\left(\frac{1}{22}\sum_i \ln(1+s_i)\right) - 1$, where s_i is the final-window percentage for achievement i defined above. This definition aggregates each achievement's rate before taking the geometric mean. Because s_i is measured in percentage points, the unit offset behaves as a one-percentage-point pseudocount for each achievement, softening the penalty for near-zero rates and compressing differences at the bottom of the table. The original one-million-interaction benchmark forms s_i from all training episodes. Our billion-interaction

Achievement	CALOR	C-TeC	E3B	ICM	RND	Alive
collect_wood	99.99	99.55	99.87	91.17	35.23	43.74
place_table	99.79	99.09	98.37	79.46	5.54	21.30
make_wood_pickaxe	99.39	36.90	14.50	6.56	0.28	6.75
collect_stone	98.50	1.63	1.94	0.20	0.02	1.38
eat_cow	98.11	58.25	0.49	35.14	13.91	93.79
defeat_zombie	97.36	0.85	0.17	12.56	31.69	1.84
wake_up	97.08	0.00	95.07	11.35	33.74	98.34
place_stone	97.01	0.08	0.86	0.01	0.01	0.71
collect_coal	95.91	0.10	0.12	0.01	0.00	0.14
collect_drink	95.72	99.61	12.39	78.85	40.56	96.16
make_stone_sword	91.45	0.00	0.03	0.00	0.00	0.04
make_wood_sword	89.93	47.63	14.44	6.05	0.27	10.88
make_stone_pickaxe	84.53	0.01	0.03	0.00	0.00	0.02
place_furnace	74.27	0.07	0.87	0.01	0.01	0.66
collect_iron	70.92	0.00	0.00	0.00	0.00	0.00
collect_sapling	70.32	4.58	23.85	24.52	6.00	46.03
defeat_skeleton	65.96	0.03	0.02	0.20	0.08	0.27
place_plant	32.64	3.12	17.20	23.54	4.06	42.86
eat_plant	5.78	0.00	0.00	0.00	0.00	7.57
make_iron_pickaxe	0.32	0.00	0.00	0.00	0.00	0.00
make_iron_sword	0.24	0.00	0.00	0.00	0.00	0.00
collect_diamond	0.00	0.00	0.00	0.00	0.00	0.00

Table 3: **Per-achievement success rates.** Per-achievement success rates (percent of episodes) on reward-free Craftax-Classic at one billion interactions. Rows are ordered by CALOR's rate.

learning curves instead use a trailing window to describe the behavior reached at each training point rather than its cumulative history. This choice prioritizes final results over early performance.

Return against score. The two aggregate metrics disagree about the baselines. Ranked by final episode return they order C-TeC (3.61), E3B (2.90), ICM (2.80), and ranked by Crafter score they order ICM, E3B, C-TeC, an exact inversion. Return rewards survival and repetition while the score rewards breadth of achievement, and the three baselines trade these off differently. CALOR leads on both at 14.76 ± 0.31 return and 40.7 ± 2.7 score, so no conclusion in this paper depends on which is chosen, but the inversion is a reason to report both. Figures 9 and 10 give the return and episode-length curves. All methods share the same maximum episode length of $10{,}000$ steps. At the end of training the baseline means range from 141 to 391 steps, while Alive reaches 826 and CALOR reaches 2144, with individual CALOR episodes reaching the cap. The achievement rates in Table 3 suggest a survival bottleneck rather than different truncation. Alive and CALOR exceed 90% on `eat_cow`, `collect_drink`, and `wake_up`, while each baseline is weak on at least one of these sources of food, water, or restored energy. For example, C-TeC has a 0.00% `wake_up` rate, E3B has a 0.49% `eat_cow` rate, ICM has an 11.35% `wake_up` rate, and RND has a 13.91% `eat_cow` rate.

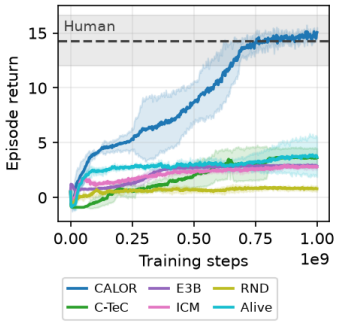


Figure 9: **Craftax-Classic episode return.**

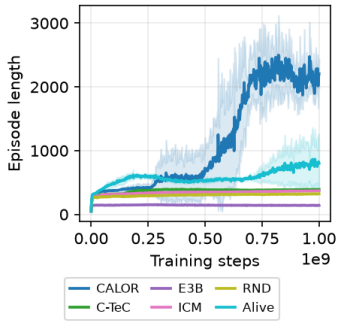


Figure 10: **Craftax-Classic episode length.**

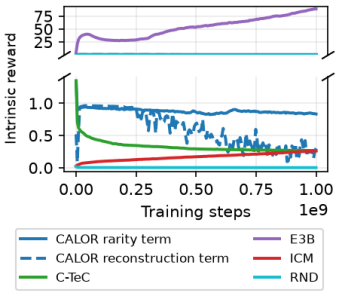


Figure 11: **Craftax-Classic intrinsic reward.**

Method	Mean intrinsic reward	Episode length
CALOR (rarity term)	0.88	
CALOR (reconstruction term)	0.59	2144
CALOR (total)	1.47	
C-TeC	0.33	391
E3B	53.4	141
ICM	0.17	369
RND	0.0039	322
Constant survival control	1.5	826

Table 4: **Craftax-Classic intrinsic reward and episode length.** Mean intrinsic reward overall and mean episode length at the end of training.

Intrinsic reward drift. Figure 11 plots the mean intrinsic reward of each method over training and Table 4 shows the overall mean intrinsic reward and episode length. The decline of CALOR's reconstruction term is by design. Normalizing by the square root of the second moment of reconstruction error, rather than by its mean, keeps the root mean square reward near one without forcing its mean to remain constant. As the codec masters more observations, their errors gather near zero while the remaining errors become sparser, so the first moment falls relative to the second. Mastered observations therefore approach zero reward while the less familiar observations retain a reward of order one. The slow EWMA decay of 0.999 keeps this reference scale stable over time.

Continuous control. Table 5 gives the number of distinct states visited at the end of training on each of the three continuous-control tasks.

Method	Ant hardest maze	Humanoid U maze	Arm binpick hard (k)
CALOR	2902 ± 14	1130 ± 98	277.7 ± 7.6
C-TeC	2863 ± 30	224 ± 60	238.0 ± 86.3
APT	2125 ± 224	114 ± 10	89.1 ± 20.2
RND	2056 ± 259	186 ± 46	44.0 ± 12.4
ICM	683 ± 241	24 ± 3	33.4 ± 17.3

Table 5: **Continuous-control distinct states visited.** Distinct states visited at the end of training. Arm binpick counts are in thousands.

Table 6 lists every configuration of Sections 7 and 6 under both aggregate metrics. Figures 12–15 give the corresponding episode-return curves.

Group	Configuration	Crafter score	Episode return
	CALOR (default)	40.7 ± 2.7	14.8 ± 0.3
Reward terms	Reconstruction reward only	9.2 ± 0.4	7.3 ± 0.2
	Rarity reward only, without episode age	24.5 ± 1.5	11.6 ± 0.3
	Rarity reward only, with episode age	1.1 ± 0.1	0.8 ± 0.1
Predictor losses	No ordering loss	14.9 ± 4.8	8.7 ± 0.9
	No calibration loss	11.6 ± 2.1	7.8 ± 0.4
Ordering sources	No episode-age ordering	27.6 ± 1.0	12.3 ± 0.2
Pareto shape	$\alpha = 1$	8.1 ± 1.1	6.6 ± 0.8
	$\alpha = 3$	42.0 ± 0.8	14.9 ± 0.2
Minibatch construction	Correlated minibatches	27.8 ± 13.2	12.0 ± 2.8
Codec objective	Unweighted mean squared error	17.3 ± 6.1	9.9 ± 1.7

Table 6: **Score and return for ablations on reward-free Craftax-Classic.**

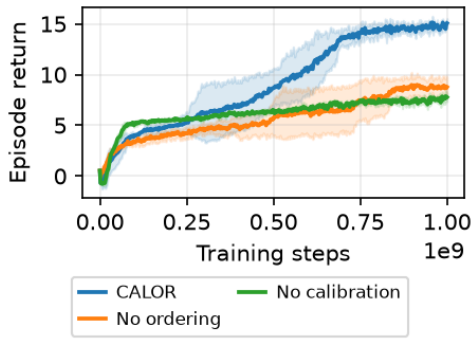


Figure 12: **Craftax-Classic episode return for the calibration and ordering ablations.**

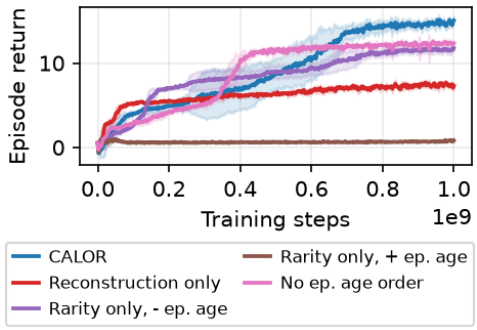


Figure 13: **Craftax-Classic episode return for the reconstruction and episode-age ablations.**

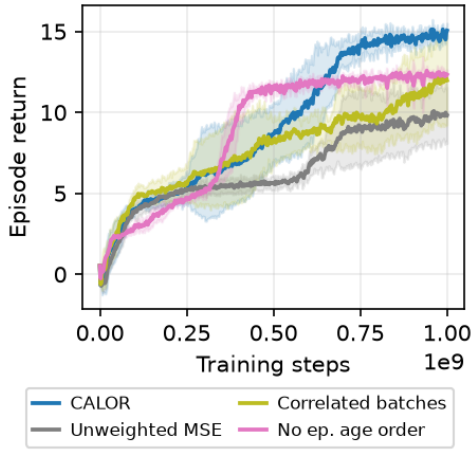


Figure 14: **Craftax-Classic episode return for the design choices of Section 6.**

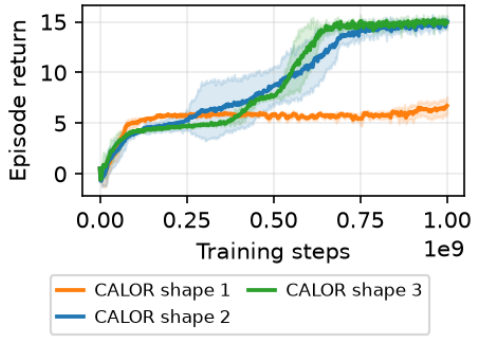


Figure 15: **Craftax-Classic episode return for three variations of Pareto shape.**