COUNTING CAPABILITIES, NOT STATES: CAPABILITY REPRESENTATIONS FOR EXPLORATION IN OPEN-ENDED WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Intrinsic exploration bonuses reward novelty, and novelty is nearly always measured over states. In open-ended environments with deep achievement hierarchies, we find that the choice of novelty space is the dominant effect. We hold the standard count bonus fixed and vary only the space it counts. Every state-derived space (pixels, hashed features, prediction error) stays near the no-bonus floor, because appearance novelty is abundant yet orthogonal to progress: an agent can sightsee forever without gaining a capability. Counting the agent's *capability configuration* instead, the binary vector of which items it holds, read from the environment's inventory for counting only, turns the same bonus into a climbing signal: it is worth $+15.7$ points on a DreamerV3-class agent that already observes its achievement state, and $+0.5$ on the same agent stripped of that input and its crafting-action prior. With that scaffold, the agent sets a new state of the art among agents trained from scratch on Crafter (67.7 vs. the prior best 58.1 at 10M steps) and raises collection of the deepest milestone, the diamond, from 0.5% to 22% of episodes. A ladder of single-flag ablations attributes every point of the improvement and uncovers a second phenomenon: without extended replay retention the agent *forgets its own breakthroughs*: a rare deep-tier trajectory leaves the default buffer within 5×10^5 steps, and holding it four times longer halves the relapses and adds 6.9 points. The same bonus lifts success on sparse-reward Meta-World pick-place from 0.22 to 0.83, and it is inert on dense-reward Meta-World and on Craftax, where the reward is already dense enough to learn from. Curiosity, in progression-structured worlds, should be about what you can do, not what you see.

1 INTRODUCTION

Crafter (Hafner, 2022) was designed to measure open-ended progress. Behind a single pixel observation sits a 22-achievement technology tree (collect wood, craft tools, smelt iron, and only then mine a diamond), and the official score is dominated by how deep into this hierarchy an agent gets. Exploration bonuses were designed for exactly this kind of challenge: intrinsic rewards for novelty (Bellemare et al., 2016; Tang et al., 2017; Pathak et al., 2017; Burda et al., 2019) exist to carry agents through long stretches where the environment pays nothing. We run the standard toolkit on an agent, budget, and configuration exactly matched to ours. Every method, from count bonuses over hashed pixels or features to prediction-error curiosity, lands within noise of *no bonus at all*, and none of them reliably crafts an iron pickaxe (Section 6). The methods built to measure novelty stall on the benchmark built to measure progress.

The reason is structural. Every method above measures novelty over *states*, and in an open world state novelty is *appearance novelty*: every new map layout, mob arrangement, and lighting change is novel in pixel space, and an agent can consume it indefinitely without ever holding a new item (Figure 1a). What is scarce, and progress-bearing, is *capability novelty*, a change in what the agent *can do* (Figure 1b). The deep end of Crafter shows the gap. The strongest agent trained from scratch, EMERALD (Burchi & Timofte, 2025), forges iron tools in 41% of episodes yet collects a diamond in only 0.5%, and methods that inject task structure fare no better unless the structure is extensive (Section 2).

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.

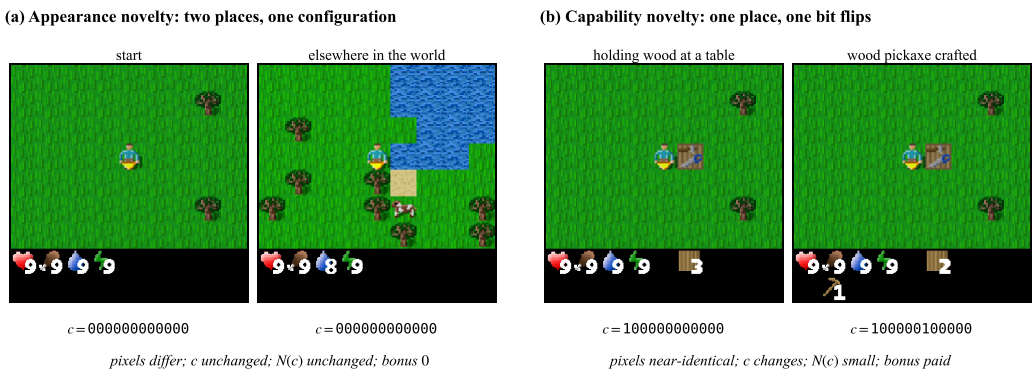


Figure 1: What the bonus counts. **(a)** Two places in a world differ in pixels but share one capability configuration c, so the count over c pays nothing for the move. **(b)** The same place before and after crafting a pickaxe differs by one inventory bit, the only change the bonus rewards. Rendered observations with the inventory set by hand; not agent trajectories.

Our proposal is a one-line change. We keep the textbook count bonus $\beta/\sqrt{N+1}$ (Bellemare et al., 2016; Tang et al., 2017) exactly as it is and change only what it counts: not a state but the agent's *capability configuration* c, a short binary vector recording which inventory items it currently holds, read from the environment for counting only. We call the result *capability-frontier exploration* (CFE), a simple form of item-keyed counting (Kanitscheider et al., 2021; Ecoffet et al., 2019; GX-Chen et al., 2025; Ferrao & Cunha, 2025) for a standard DreamerV3-class agent (Hafner et al., 2023). Because the scale, cap, injection point, and counting scheme are all held fixed, CFE can be set against hashed pixels, hashed features, and prediction error on one agent, a controlled comparison the item-keyed methods above do not make. Capability configurations change only when the agent gains or loses a capability, so their visit counts track position in the achievement hierarchy rather than appearance, and the same bonus that funded sightseeing now concentrates exploration on the *capability frontier*: configurations the agent has rarely been able to reach.

The novelty space turns out to be the lever. With the mechanism fixed, every state-derived space is worth nothing over no bonus at all, and the capability space is worth $+15.7$ points against a matched floor and opens the iron tier (Section 6). Two further single-flag additions, two workshop-adjacency bits in the counted key and longer replay retention, carry the agent to 67.7 at 10M steps, a new state of the art among agents trained from scratch against the prior best of 58.1, and raise collection of the deepest milestone, the diamond, from 0.5% to 22% of episodes (Section 5). A ladder of single-flag ablations attributes every point of the climb and shows that the gain is conditional on the agent's capability-aware perception (Section 6).

Counting capabilities also makes a falsifiable prediction: the bonus should help where a capability-gated frontier is the bottleneck, and be inert elsewhere. We test both directions (Section 7). On sparse-reward Meta-World pick-place (Yu et al., 2019) the bonus turns a task the base agent never solves into one it solves in most episodes, and it is inert once the reward is dense. On sparse-reward Franka Kitchen (Gupta et al., 2019), a manipulation domain with no technology tree, the same bonus over a capability vector of completed subtasks enlarges the configurations reached in every matched seed but leaves the task score unresolved at two budgets. On Craftax (Matthews et al., 2024), whose reward is already dense enough to learn from, it is inert, as the prediction says.

2 Related Work

Novelty bonuses over states. Rewarding rarely visited states is a classical exploration principle, extended to high dimensions by density-model pseudo-counts (Bellemare et al., 2016) and by counting a fixed hash of the observation (Tang et al., 2017). Later bonuses refine how novelty is estimated rather than what it is measured over: ICM (Pathak et al., 2017) rewards forward-model prediction error, RND (Burda et al., 2019) the error of predicting a random network, E3B (Henaff et al., 2022) an episodic elliptical bonus, and others add episodic and lifelong terms (Badia et al., 2020), change

in a learned embedding (Raileanu & Rocktäschel, 2020), novelty differences between consecutive states (Zhang et al., 2021), or an adversarial critic (Flet-Berliac et al., 2021). World-model agents can also explore through their own model, seeking ensemble disagreement in imagination (Sekar et al., 2020), imagined goals (Mendonca et al., 2021), or a learned hierarchy of latent goals (Hafner et al., 2022). These change the exploration objective and the agent; our comparison holds both fixed and varies only the counted space, so they fall outside it.

Every one of these measures novelty over observations, learned features, or the latent state. The design choice in this family is what is counted; hashing observations spreads novelty over appearance, and Section 6 compares that choice directly against counting a capability abstraction, with the bonus mechanism held fixed and RND as the learned-novelty comparator.

Counting over items and capabilities. Item-keyed novelty has precedents, mostly in Minecraft, where diamonds have otherwise been reached with human data (MineRL demonstrations (Guss et al., 2019), VPT's contractor play (Baker et al., 2022), Voyager's language-model scripting (Wang et al., 2023)) or, from sparse reward alone, by DreamerV3 with abstract crafting actions (Hafner et al., 2023). Kanitscheider et al. (2021) pay a within-episode bonus for each item first obtained in Minecraft, restricted dynamically to items the agent cannot yet reliably get; Go-Explore (Ecoffet et al., 2019) records held keys in the cells whose visit counts steer its archive; MEAD (GX-Chen et al., 2025) counts (item, attribute) abstract states, "in inventory" among them, to plan exploration in 2D crafting and MiniHack; CBET (Ferrao & Cunha, 2025) counts hashed states and state changes with DreamerV3 on Crafter. In contrast to these, CFE leaves the count bonus unchanged, counts a thresholded inventory vector at data-collection time, and runs on a world-model agent. None of them holds the bonus rule fixed and swaps only the counted space against state-derived spaces on one agent, and none reports Crafter's deep tier.

Agents on Crafter. Progress on Crafter has come from two directions. One adds task structure. SEA (Zhou & Garg, 2023) recovers the achievement-dependency graph and trains an offline achievement classifier, reaching the diamond ($4.21\% \pm 3.24\%$) under a much larger, modified-environment regime; Achievement Distillation (Moon et al., 2023) distills achievement-keyed behavior and, at its 1M-step budget, reports iron tools in $\approx 0.01\%$ of episodes and diamonds only on isolated runs; ELLM (Du et al., 2023) imports a language-model goal prior, and its Crafter evaluation targets pretraining coverage and downstream tasks rather than the achievement score, so it reports no deep-tier rate (Table 6); object-centric architectures have been studied for generalization (Stanić et al., 2022).

The other strengthens the world model. The Dreamer line (Hafner et al., 2021; 2023), transformer world models (Micheli et al., 2023; Robine et al., 2023; Micheli et al., 2024), and EMERALD (Burchi & Timofte, 2025), whose spatial latent with MaskGIT prediction holds the best published score among agents trained from scratch, dominate the leaderboard; Curious Replay (Kauvar et al., 2023) instead reweights which stored experiences the world model trains on. Neither direction changes the exploration signal; CFE does, on a standard world model.

3 METHOD

Capability representation. *Capability-frontier exploration* (CFE) computes novelty over a *capability representation* c, a summary of what the agent can currently do, with three properties. **(P1) Low cardinality:** c ranges over hundreds of reachable values, not an exponential state space, so visit counts are meaningful. **(P2) Capability-gated:** c changes only when the set of things the agent can do changes, never on appearance, position, or time. **(P3) Persistent:** acquired capabilities tend to be kept within a life, so c advances approximately monotonically and its counts track progression rather than revisits. Consumables flip back when spent (wood into a pickaxe, Figure 3), but they flip into a configuration that holds the tool, itself new, and back to ones already well counted. Observation-derived spaces violate all three, and Section 6 shows that the properties predict which novelty signals work. In this paper c is read from exposed state; Section 8 reports a first attempt to learn it. The comparison arms of Section 6 differ from CFE only in the space the bonus counts (Figure 1).

Capability configuration. At each environment step we form the capability configuration

$$c_t \;=\; \big(\mathbb{1}[\text{holds item } 1],\; \ldots,\; \mathbb{1}[\text{holds item } k]\big) \in \{0,1\}^k,$$

one indicator per inventory item type, thresholded at *holds at least one* (Crafter's twelve items: sapling, wood, stone, coal, iron, diamond, and the wood, stone, and iron pickaxes and swords; the four vital meters are not counted). Thresholding collapses inventory quantities into a small set of capability states whose transitions are the acquisition events of the technology tree (first wood, first iron), without naming or ordering them. Optionally, coarse situational flags that relax P2 can be appended to c; on Crafter we study two privileged workshop-adjacency bits (a crafting table, respectively a furnace, in the 3×3 neighbourhood of the agent, whatever it faces) and measure their contribution separately ($+4.1$; Section 6). The configuration is computed from the environment's exposed inventory state for *counting only*; it is not an input to the agent's networks.

Bonus. A table of visit counts $N(c)$ is maintained, incremented once per real environment step, and the intrinsic reward is the textbook count bonus with a fixed scale and cap,

$$r_t^{\text{int}} = \min\Big(\beta/\sqrt{N(c_t)+1},\ 0.5\Big), \qquad r_t = r_t^{\text{env}} + \lambda\, r_t^{\text{int}}, \tag{1}$$

with β=1 and λ=0.5 in all experiments. Counting must happen at data-collection time: counting inside the learner's update loop inflates N by the replay-resample factor (≈ 128 here) and damps the bonus by $\approx 11\times$. The bonus self-anneals (mastered configurations pay vanishing bonus, shifting exploration to the frontier) and is removed at evaluation; all reported scores use the true environment reward only.

Base agent. We use `r2dreamer`, a decoder-free DreamerV3-class agent whose representation is learned with R2-Dreamer's redundancy-reduction objective (Morihira et al., 2026), a Barlow Twins-style loss (Zbontar et al., 2021). Following its Crafter recipe, the agent carries an auxiliary achievement-prediction task (the 22-dim achievement state as an encoder input and a BCE prediction head) and an exploration ϵ-prior that, with probability 0.1 during training, resamples the action uniformly from the crafting/placing actions. Every auxiliary component is ablated with single-flag removals in Section 6: removing the prediction head does not change the score, and the remaining components are quantified in Table 4. The environment's reward and dynamics are unmodified, and the agent receives no reward shaping, demonstrations, or privileged resets.

4 Experimental Setup

Benchmark and metric. Original Crafter (Hafner, 2022) with the standard reward; official score $= \exp\left(\frac{1}{22}\sum_i \ln(1+s_i)\right) - 1$ over the 22 achievement success rates s_i (in percent), evaluated with the standard 256-episode protocol. **Budget and seeds.** 10M environment steps (EMERALD's budget); 6M for sample-efficiency analysis. Three seeds per configuration unless noted; per-seed values in Appendix A. **Default configuration.** Unless otherwise specified, CFE denotes rung 2 of Table 2 (count bonus over the inventory and workshop bits, replay buffer 5×10^5). Every "bonus off" value is the configuration named beside it with the bonus switched off and nothing else changed. With three to six seeds per cell, exact tests reach only their smallest attainable p-values; we apply no multiplicity correction. **Baselines.** Leaderboard rows are quoted from the literature. Mechanism comparisons (SimHash on pixels, SimHash on features, RND on the real-reward pathway, bonus-off) run on the identical agent, budget, and configuration; only the exploration signal differs. Our reproduction of EMERALD from the authors' code reaches 58.7 on its best seed and 46.9 ± 8.5 over five (one seed evaluated at epoch 47 of 50 after plateauing); the authors' release reports 57.0 over 20 seeds with a 0.9% diamond rate. We compare against the published 58.1 throughout. To verify the result is not specific to the redundancy-reduction representation, CFE is re-run with the vanilla reconstruction-based DreamerV3 representation: it reaches 60.4 ± 0.9 (n=3), matching the default agent's 60.8 and confirming the gain does not depend on the redundancy-reduction objective.

5 Results

A new state of the art among agents trained from scratch. Table 1 places CFE on the 10M Crafter leaderboard. Among agents trained from scratch, CFE with a $4\times$ replay buffer (retention) reaches 67.7 ± 2.0 over three seeds and 67.3 ± 3.3 over five (Appendix A), 9.6 points above the previous best, EMERALD at 58.1. The default buffer gives 60.8 ± 0.6, every seed above 58.1. It also

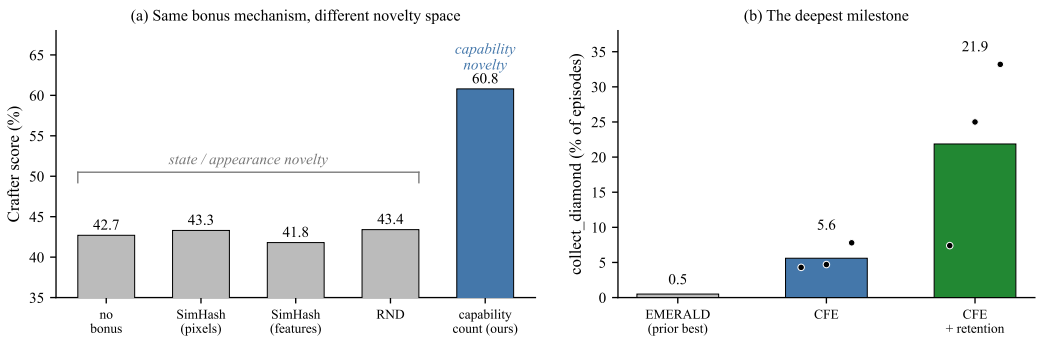


Figure 2: **(a)** Counting the capability configuration is worth $+18$ over the no-bonus arm of this comparison (the rung-2 configuration of Table 2); every state or appearance space lands within noise of no bonus. Same count-bonus mechanism (envelope, scale, cap, injection point), same agent and budget. **(b)** Diamond collection rises from 0.5% (prior best) to 5.6%, and to 21.9% with retention (dots are seeds).

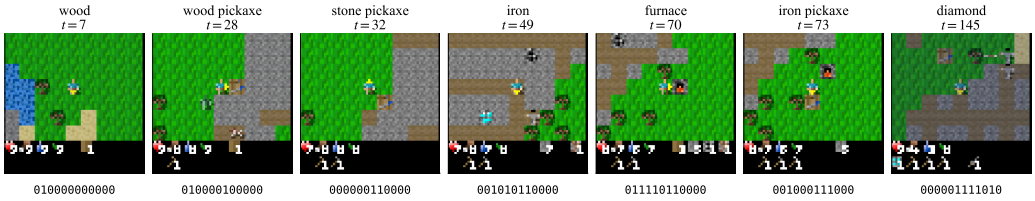


Figure 3: One evaluation episode, shown at the step each capability on the diamond chain is first acquired; the diamond is collected at $t = 145$. Frames are the agent's 64×64 observations with the twelve inventory bits of c below each (the two workshop bits are not shown). Rung-2 configuration of Table 2, 4.1M-step checkpoint.

exceeds C-VPT (Park et al., 2025) (61.4 ± 4.7), which is behavior-cloned from 20,000 demonstrations of a scripted expert that itself scores 97.5. The exposed inventory and workshop state enters only the counting bonus, never the agent's networks (Section 3); the base agent's own achievement input is ablated in Table 4, and with the bonus off the same configuration scores 42.7 (Table 3).

Deep-tier achievements. Figure 2(b) decomposes the score at the deepest milestone (per-achievement detail in Appendix A). Agents trained from scratch had already closed the iron-tool gap (EMERALD: 41.5% iron pickaxe; CFE: 45–73%); the diamond was the open frontier, and that is where CFE moves the needle: $0.5\% \rightarrow 5.6\%$ at the default configuration and 22% with retention, a $44\times$ increase at the single hardest milestone (20.5%, or $41\times$, across all five seeds). The default rate already exceeds SEA's structured 4.21%, obtained with a recovered achievement graph and $\sim 50\times$ compute on a modified environment. The gain therefore comes from reaching a deeper milestone rather than from collecting more of the shallow ones: a generic curiosity signal reaches a milestone that a stronger world-model architecture does not.

6 Analysis: Where the Improvement Comes From

We attribute the entire improvement with a ladder of single-flag ablations (Table 2; the same rungs are plotted in Figure 5). Starting from a floor configuration that is identical to CFE except for the bonus scale (λ=0), each rung changes one flag; each is run with three seeds.

Novelty space. Turning on the bonus over the inventory-only key, one flag (rung 0 to rung 1 of Table 2), is worth 15.7 points and opens the iron tier (iron pickaxe $0\% \rightarrow 41\%$). To locate the gain in the counted space rather than the bonus rule, we hold the whole mechanism fixed (the same $\beta/\sqrt{N+1}$ envelope, scale, cap, and real-reward injection) and change only the space (Figure 2a;

Table 1: Crafter at 10M steps, official 256-episode protocol: score and success rate of the deepest achievement. †Published values as tabulated by Burchi & Timofte (2025). Our rows are mean ± sd over three seeds; the EMERALD reproduction runs the authors' code over five (Section 4), and margins in the text use the published 58.1. Among agents trained from scratch, the best value per column is in **bold** and the second best is underlined. n/r: not reported.

Method	Score (%)	Diamond (%)	Params
Human experts† (Hafner, 2022)	50.5	12.0	—
From scratch: pixels and reward†			
DreamerV3-M (Hafner et al., 2023)	34.9	0.0	37M
DreamerV3-XL (Hafner et al., 2023)	39.6	0.0	200M
Δ-IRIS (Micheli et al., 2024)	42.5	0.0	25M
EMERALD (Burchi & Timofte, 2025)	<u>58.1</u>	<u>0.5</u>	30M
our reproduction	46.9 ±8.5	—	30M
From scratch: reward and exposed game state (ours)			
CFE	60.8 ±0.6	5.6	∼50M
CFE + retention	**67.7** ±2.0	**21.9**	∼50M
Imitation of a scripted expert			
C-VPT-large (Park et al., 2025)	61.4 ±4.7	n/r	64M
its scripted expert policy	97.5	n/r	—

Table 2: The decomposition ladder (Crafter score %, 10M steps, n=3 per rung). Each rung differs from the previous by one configuration flag.

#	Configuration	Score	Δ	Mechanism
0	rung 1 with the bonus off	41.0 ± 0.3	—	iron/diamond 0%
1	+ count bonus, inventory-only key	56.7 ± 1.2	+15.7	the capability abstraction
2	+ workshop bits in the key	60.8 ± 0.6	+4.1	privileged obs; deepest tier
3	+ retention (4× buffer)	67.7 ± 2.0	+6.9	amplifies the bonus (Table 3)

Table 7). SimHash counts over pixels or over the agent's latent features score 43.3 and 41.8 against 42.7 for the default configuration with the bonus off (Table 3). Counting the same hashed features at collection time, the schedule of Section 3, leaves the arm at the floor as well (39.9 ± 1.8, n=3), so the null is in the space and not in the schedule; by 1M steps the 2^{14} hash buckets are nearly exhausted on every seed, as P1 predicts. RND lands at 43.4 ± 4.0, with one of its five seeds at the iron tier and four at the floor. Counting the capability configuration on the same agent scores 60.8. This is what P1–P3 predict: a hashed observation space has millions of buckets that advance on every camera pan, so its novelty is spent on sightseeing, while the capability space changes only when a capability does, so the same bonus is spent on the frontier.

Workshop bits. Appending the two workshop-adjacency bits to the counted key adds 4.1 points, concentrated in the deepest tier (diamond 0.9% → 5.6%; iron sword 21% → 43%). They are the only privileged bits in the counted space; the rest of the bonus effect comes from generic inventory state.

Replay retention. Crossing the bonus with the buffer size (Table 3) shows that the 4× buffer is worth +6.9 with the bonus on and +2.8 without it (n=2 per bonus-off cell). A first diamond is a once-in-many-episodes event, and at the default buffer size its trajectory leaves replay within 5×10^5 steps, after which the breakthrough has to be rediscovered. The evaluation stream shows this: the default-buffer arm loses a deep-tier achievement for at least 100k steps twice as often as the 4× arm (16 vs. 8 relapses over three seeds and three achievements; Appendix A.3). Under uniform sampling every transition is trained on the same expected number of times whatever the buffer size, so the larger buffer does not train on a

	Buffer 5×10^5	**Buffer** 2×10^6
Bonus on	60.8 ± 0.6	67.7 ± 2.0
Bonus off	42.7	45.5

Table 3: Bonus × buffer at 10M (Crafter score %), default configuration with the bonus on or off. Bonus-on cells n=3, bonus-off n=2.

Table 4: Single-flag removals of base-agent components (10M, n=3 unless noted): Crafter score and the success rates of the two deepest achievements, official 256-episode evaluation, mean over seeds. No single removal collapses the gain; it collapses only when all are removed.

Configuration	Score (%)	Iron pick. (%)	Diam. (%)	Removes
CFE (rung 2)	60.8 ± 0.6	45.4	5.6	—
− achievement head	63.4 (n=2)	46.9	7.5	auxiliary BCE head
− workshop bits (rung 1)	56.7 ± 1.2	41.0	0.9	privileged obs
rung 1 − achievement observation	47.0 ± 5.7	16.0	1.7	encoder achievement input
same, no bonus	40.5 ± 0.4	0.0	0.0	its matched floor
rung 1 − crafting ϵ-prior	47.5 ± 7.2	9.2	0.0	action prior
same, no bonus	42.8 ± 1.3	0.3	0.0	its matched floor
minimal (bonus only)	40.5 ± 0.4	0.1	0.0	all of the above
minimal, no bonus (its floor)	40.0 ± 0.3	0.1	0.0	—

breakthrough more often; it keeps the breakthrough available four times longer while the policy moves on. Curious Replay (Kauvar et al., 2023) acts on the other factor, raising the sampling probability of steps at which a newly discovered resource is held (up to 6.2× for iron) at fixed capacity; the two are complementary and we did not combine them. Retention alone does not approach the bonus's effect: the 4× buffer with the bonus off scores 45.5 against 60.8 for the default buffer with the bonus on. Retention also reaches the default configuration's 10M score by 6M steps (61.2 vs. 60.8).

Base-agent components. Table 4 removes the base agent's auxiliary components one flag at a time from the first two rungs of Table 2. Removing the achievement-prediction head does not change the score (63.4 at n=2, slightly above the full configuration); removing the achievement observation or the crafting-action ϵ-prior is quantified in the table; the workshop bits are the +4.1 rung above.

7 When Does Capability Counting Help?

Counting capabilities makes a sharp prediction: the bonus pays where a capability-gated frontier is reachable and under-rewarded, and is inert elsewhere. We test the prediction in both directions.

Franka Kitchen. The Franka Kitchen task (Gupta et al., 2019) has no technology tree; its capability configuration is the set of completed appliance subtasks (a seven-bit vector over the microwave, kettle, two burners, light switch, and two cabinets), and the score is how many of the seven an episode completes. The bonus arm is ahead on average at both budgets we ran (+0.79 at 1M steps, +0.59 at 3M), in every matched pair at 1M, but five seeds per arm cannot resolve differences of that size (Appendix C). The bonus does enlarge coverage. Against a matched control that maintains the count table but never adds it to the reward, the bonus arm visits about twice as many capability configurations by the end of training (52 vs. 23). This holds in all five matched pairs (sign-flip $p = 0.0625$, the floor for five pairs), and the two arms' task scores are indistinguishable. On this domain the mechanism does what it claims, and whether the enlargement converts into score is unresolved at this sample size: the score differences are directionally consistent across seeds but would need far more than five per arm to settle (full statistics in Appendix C).

Meta-World pick-place. Franka Kitchen shows the enlargement without a resolved score gain; Meta-World pick-place (Yu et al., 2019) shows the other outcome.

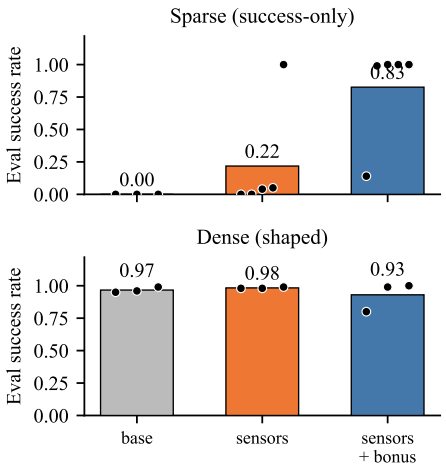


Figure 4: Meta-World pick-place. Under sparse reward the bonus arm reaches 0.83 success against 0.22 for sensors-only and 0.00 for the base agent; under dense reward all arms saturate. Dots are seeds (n=5, 5, 3 sparse; 3 dense); values are means of the final ten in-training evaluations.

Here c is the grasp and success indicators together with coarse approach bands, a deliberate relaxation of P2 for a continuous task with no discrete items: with a fixed-base arm, the band the gripper occupies is what it can do to the object next (edges in Appendix C). Under the sparse, success-only reward the bonus arm reaches 0.83 evaluation success, a matched arm given the same exposed sensors but no bonus 0.22, and the base agent 0.00 (n=5, 5, 3; per-seed values and tests in Appendix C). The bonus is worth $+0.61$ and the sensors alone $+0.22$, the same bonus-not-scaffold split as on Crafter (Table 4). Under the dense shaped reward all three arms saturate (0.97, 0.98, 0.93 for base, sensors-only, and bonus; n=3), so the bonus is inert where the environment's own reward already prices progress (Figure 4).

A second agent. We repeat the comparison on EMERALD, a pixels-only agent, as a 2×2 of achievement input and bonus (Table 5). The bonus adds $+0.6$ on the pixels-only agent, and the achievement input lowers the score with or without the bonus ($52.9 \to 42.0$ and $53.5 \to 47.9$). The interaction is $+5.3$ (SE 6.9; 95% CI $[-8.5, +19.0]$), which covers both zero and the Crafter-sized $+15.7$, so this replication neither reproduces nor rules out the conditionality (Appendix A.2).

Craftax. On Craftax (the full game, 65 achievements) the bonus is null at the 1M budget its authors propose, ported into the official PPO baseline on Craftax-Symbolic-v1 (n=5, $+0.02$ percentage points), and within seed noise at 10M with our own agent on Craftax-Pixels-v1 (7.65 vs. 7.62, two seeds per arm, both reaching the same 22 achievements). The benchmark's own study finds the same for RND, ICM, and E3B and attributes it to a reward structure already dense enough to learn from (Matthews et al., 2024); by 10M the count bonus has also fully annealed (257 configurations visited). Craftax has no under-rewarded capability frontier for the bonus to find (Appendix C).

	no bonus	bonus (CFE)
pixels only	52.9 ± 11.4 (5)	53.5 ± 6.0 (6)
+ achievement input	42.0 ± 0.2 (4)	47.9 ± 7.5 (5)

Table 5: EMERALD 2×2 at 10M steps (Crafter score, official 256-episode evaluation; mean $\pm$ sd over clean seeds, n in parentheses; resumed seeds excluded).

8 Discussion

Learning the capability representation. A learned capability key does not yet work as a count key (Appendix A.1). Nothing in P1–P3 requires reading c from exposed state, so we learn discrete bits over the world model's latent state, ratcheted monotone within a life and trained on capability-change events rather than labels. A purely learned key reaches 37.8 ± 2.1 (n=3, 10M) and a hybrid key that adds the explicit inventory bits 37.8 ± 3.7, both at the no-bonus floor and far below the 56.7 the explicit bits earn alone; neither opens the iron tier. Offline probes nonetheless recover the full 22-achievement tree from the learned bits (per-achievement AUC 0.94–0.95). The failure is in the coupling, possibly because counting over noisy learned bits feeds that noise back into the reward predictor the policy relies on; shrinking the bonus or the code does not remove it. A usable learned key therefore needs a different bonus coupling.

Limitations. The capability configuration is specified per domain from exposed state (inventory on Crafter; inventory and dungeon floor on Craftax; completed appliance subtasks on Franka Kitchen; grasp/success gates and coarse approach bands on Meta-World): trivial where such state exists, undefined from raw pixels alone. The learned-representation experiments above measure what replacing it currently requires. Learned novelty over appearance needs none of this and does not open the deep tier in our controlled comparison (Section 6); the learned capability key of Appendix A.1 does not yet either.

9 Conclusion

Exploration bonuses have always asked "how novel is this?", and have nearly always answered by looking at the state. In environments with deep achievement hierarchies, the better question is "how novel is what I can do?" Asking it, by counting the agent's capability configuration with an otherwise

textbook count bonus, takes a standard world-model agent past the best published from-scratch Crafter score and turns the benchmark's deepest milestone, the diamond, from a rare event into a routine one. Where the environment exposes what the agent holds, counting capabilities rather than states is a one-line change to a standard bonus. Whether the configuration can be learned rather than read is the question our negative results leave open, and the one we think matters most.

Reproducibility Statement

All arms run from a single codebase and differ only in the configuration flags named in the text. The complete flag set and commit for each run will be released with the code, alongside per-seed scores (Appendix A) and training logs. The bonus is fully specified by Eq. (1) with β=1, λ=0.5, cap 0.5 in all experiments, counted once per real environment step at data-collection time (Section 3); beyond that bonus no run uses reward shaping, demonstrations, or privileged resets. Crafter scores follow the official 256-episode protocol at 10M environment steps. Franka Kitchen is scored separately, as the mean number of seven appliance subtasks completed over 100 evaluation episodes per seed, using the environment's own completion signal; capability-configuration coverage is read from training logs rather than from an evaluation protocol. Seed counts are stated per table. The EMERALD comparison uses the authors' published number at the matched budget, and our reproduction from their code is reported in Section 4.

AI Use Statement

We used a generative AI assistant to read the manuscript, provide feedback on clarity and presentation, and suggest edits to individual sentences, which we reviewed and incorporated at our discretion. Generative AI was not used to design experiments, produce or analyze experimental results, or draft the paper's content. We take responsibility for the final content of this work.

References

A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, and C. Blundell. Never give up: Learning directed exploration strategies. *International Conference on Learning Representations (ICLR)*, 2020. arXiv:2002.06038.

B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video PreTraining (VPT): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2206.11795.

M. G. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. arXiv:1606.01868.

M. Burchi and R. Timofte. Accurate and efficient world modeling with masked latent transformers (EMERALD). *International Conference on Machine Learning (ICML)*, 2025. arXiv:2507.04075.

Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1810.12894.

Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, and J. Andreas. Guiding pre-training in reinforcement learning with large language models (ELLM). *International Conference on Machine Learning (ICML)*, 2023. arXiv:2302.06692.

A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune. Go-Explore: A new approach for hard-exploration problems. *arXiv preprint*, 2019. arXiv:1901.10995.

J. Ferrao and R. Cunha. World model agents with change-based intrinsic motivation (CBET). *arXiv preprint*, 2025. arXiv:2503.21047.

Y. Flet-Berliac, J. Ferret, O. Pietquin, P. Preux, and M. Geist. Adversarially guided actor-critic (AGAC). *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2102.04376.

A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning (CoRL)*, 2019. arXiv:1910.11956.

W. H. Guss, C. Codel, K. Hofmann, B. Houghton, N. Kuno, S. Milani, S. Mohanty, D. Perez Liebana, R. Salakhutdinov, N. Topin, M. Veloso, and P. Wang. The MineRL 2019 competition on sample efficient reinforcement learning using human priors. *NeurIPS 2019 Competition Track*, 2019. arXiv:1904.10079.

A. GX-Chen, K. Marino, and R. Fergus. Efficient exploration and discriminative world model learning with an object-centric abstraction (MEAD). *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.11816.

D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering Atari with discrete world models (DreamerV2). *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.02193.

D. Hafner. Benchmarking the spectrum of agent capabilities (Crafter). *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2109.06780.

D. Hafner, K.-H. Lee, I. Fischer, and P. Abbeel. Deep hierarchical planning from pixels (Director). *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2206.04114.

D. Hafner, J. Pašukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models (DreamerV3). *arXiv preprint*, 2023. arXiv:2301.04104.

M. Henaff, R. Raileanu, M. Jiang, and T. Rocktäschel. Exploration via elliptical episodic bonuses (E3B). *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2210.05805.

I. Kauvar, C. Doyle, L. Zhou, and N. Haber. Curious replay for model-based adaptation. *International Conference on Machine Learning (ICML)*, 2023. arXiv:2306.15934.

I. Kanitscheider, J. Huizinga, D. Farhi, W. H. Guss, B. Houghton, R. Sampedro, P. Zhokhov, B. Baker, A. Ecoffet, J. Tang, O. Klimov, and J. Clune. Multi-task curriculum learning in a complex, visual, hard-exploration domain: Minecraft. *arXiv preprint*, 2021. arXiv:2106.14876.

M. Matthews, M. Beukman, B. Ellis, M. Samvelyan, M. Jackson, S. Coward, and J. Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.16801.

R. Mendonca, O. Rybkin, K. Daniilidis, D. Hafner, and D. Pathak. Discovering and achieving goals via world models (LEXA). *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2110.09514.

V. Micheli, E. Alonso, and F. Fleuret. Transformers are sample-efficient world models (IRIS). *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.00588.

V. Micheli, E. Alonso, and F. Fleuret. Efficient world models with context-aware tokenization (Δ-IRIS). *International Conference on Machine Learning (ICML)*, 2024. arXiv:2406.19320.

S. Moon, J. Yeom, B. Park, and H. O. Song. Discovering hierarchical achievements in reinforcement learning via contrastive learning (Achievement Distillation). *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2307.03486.

N. Morihira, A. Nahar, K. Bharadwaj, Y. Kato, A. Hayashi, and T. Harada. R2-Dreamer: Redundancy-reduced world models without decoders or augmentation. *International Conference on Learning Representations (ICLR)*, 2026. OpenReview:Je2QqXrcQq.

J. Park, H. Cho, and S. Ahn. CrafterDojo: A suite of foundation models for building open-ended embodied agents in Crafter. *arXiv preprint*, 2025. arXiv:2508.13530.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction (ICM). *International Conference on Machine Learning (ICML)*, 2017. arXiv:1705.05363.

R. Raileanu and T. Rocktäschel. RIDE: Rewarding impact-driven exploration for procedurally-generated environments. *International Conference on Learning Representations (ICLR)*, 2020. arXiv:2002.12292.

J. Robine, M. Höftmann, T. Uelwer, and S. Harmeling. Transformer-based world models are happy with 100k interactions (TWM). *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2303.07109.

R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models (Plan2Explore). *International Conference on Machine Learning (ICML)*, 2020. arXiv:2005.05960.

A. Stanić, Y. Tang, D. Ha, and J. Schmidhuber. Learning to generalize with object-centric agents in the open world survival game Crafter. *arXiv preprint*, 2022. arXiv:2208.03374.

H. Tang, R. Houthooft, D. Foote, A. Stooke, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1611.04717.

G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint*, 2023. arXiv:2305.16291.

D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning (DrQ-v2). *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2107.09645.

T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman, C. Finn, and S. Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. *Conference on Robot Learning (CoRL)*, 2019. arXiv:1910.10897.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow Twins: Self-supervised learning via redundancy reduction. *International Conference on Machine Learning (ICML)*, 2021. arXiv:2103.03230.

T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian. NovelD: A simple yet effective exploration criterion. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Z. Zhou and S. Garg. Learning achievement structure for structured exploration in domains with sparse reward (SEA). *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2305.00508.

A PER-SEED VALUES

CFE +retention (10M): $69.88/65.86/67.26 \rightarrow 67.7 \pm 2.0$ (n=3); diamond $33.2/7.4/25.0\%$; iron pickaxe $79.7/64.8/74.6\%$. Two additional seeds (70.98/62.71; diamond $33.6/3.1\%$; iron pickaxe $84.0/59.4\%$) give 67.3 ± 3.3 at n=5, confirming the headline. The diamond rate is reported on the same three headline seeds as the score (21.9%); across all five it is 20.5%, and the per-seed spread is wide in both directions (3.1–33.6%), so the deepest-tier rate should be read as a rate with large seed variance rather than a point estimate. **CFE (rung 2):** $61.36/60.11/60.84 \rightarrow 60.8 \pm 0.6$; diamond $4.3/4.7/7.8\%$. **Bonus-off cells of Table 3:** buffer 5×10^5: $40.97/44.48 \rightarrow 42.7$; buffer 2×10^6: $49.75/41.34 \rightarrow 45.5$ (n=2 each). **Rung 2 − achievement head:** $60.26/66.56 \rightarrow 63.4$ (n=2). **Rung 1 (inventory-only key):** $57.18/57.55/55.34 \rightarrow 56.7 \pm 1.2$; iron pickaxe 41%; diamond 0.9%. **Rung 0 (matched floor):** $41.35/40.70/41.00 \rightarrow 41.0 \pm 0.3$; iron pickaxe 0%. **EMERALD reproduction (n=5):** $58.71/53.01/41.73/41.16/39.66 \rightarrow 46.9 \pm 8.5$. **Franka Kitchen (sparse, mean of 7 subtasks completed over 100 eval episodes; n=5 per arm at 1M, n=6 at 3M).** At 1M: CFE $4.25/2.93/5.28/4.80/2.68 \rightarrow 3.99 \pm 1.14$; no bonus $3.15/1.79/4.52/4.77/1.75 \rightarrow 3.20 \pm 1.44$ (gap $+0.79$, unpaired permutation $p = 0.33$; all five matched pairs favour the bonus arm, sign-flip $p = 0.0625$). At 3M (n=6): CFE $5.81/4.34/5.85/3.93/4.90/3.42 \rightarrow 4.71 \pm 1.00$; no bonus $4.55/4.71/4.99/1.00/3.49/6.00 \rightarrow 4.12 \pm 1.73$ (gap $+0.59$, unpaired permutation $p = 0.55$; paired sign-flip $p = 0.50$, $4/6$ pairs). The sixth 3M pair moved the gap from $+1.22$ to $+0.59$: its no-bonus seed scored 6.00, the highest value anywhere in this table and equal to the attainable

maximum. The hinge cabinet is completed in 0 of 22 runs, so that maximum is 6 rather than 7. **Observe-only arm** (count table maintained, bonus never added to reward; matched seeds, 1M): score $4.64/1.87/3.90/3.92/4.02 \rightarrow 3.67 \pm 1.05$, indistinguishable from both CFE ($p = 0.56$) and no bonus ($p = 0.44$). Capability configurations visited at end of training: CFE $58/59/64/45/32 \rightarrow 52$ vs. observe-only $24/12/27/33/20 \rightarrow 23$; all five matched pairs favour CFE (sign-flip $p = 0.0625$, the minimum attainable at n=5). Measured earlier in training the same comparison gives 12 vs. 9 at 200k steps (20% of the budget; ratio 1.34, $p = 0.56$) and 36 vs. 18 at 500k (50%; ratio 1.99, $p = 0.0625$), so the bonus arm's coverage advantage grows over training rather than being present from the start. Coverage and score correlate weakly across all ten runs ($r = 0.45$, $p = 0.20$). The extra configurations largely do not survive to evaluation. Per-configuration visit counts are not logged, so whether a configuration was reached once or held has no direct test here; we use the indirect test below. Counting how many of the seven appliances an arm operates in at least one evaluation episode gives 5.0 vs. 4.6, and 4.4 vs. 3.8 at a 50% reliability threshold, with a sign-flip $p \geq 0.5$ at every threshold we checked—an underpowered null, which we do not lean on. The argument rests instead on the gap between two ratios: a $2.2\times$ gap in configurations reached during training corresponds to at most half an appliance of additional evaluation-time breadth—the bonus moves an internal quantity most of which never reaches behaviour.

The same test on Crafter comes out the other way. Comparing the matched single-flag pair that defines the abstraction rung (rung 1 of Table 2 with and without the bonus, n=3 each, 56.7 vs. 41.0) achievement by achievement, the +15.7 is concentrated in the iron tier: the no-bonus arm makes an iron pickaxe in none of its 3×256 evaluation episodes and an iron sword in none of them—exactly zero in every seed, not a mean that rounds to zero—against 41.0% and 27.2% with the bonus. Fifteen of the twenty-two achievements move the other way and six move with it, though five of the twenty-two shift by less than half a point in either direction, so the genuine regressions number about a dozen (their trajectories along the full ladder are given below). The bonus trades small reliability on already-solved achievements for entry to a tier the control never reaches at all. Where Kitchen's enlarged configuration set did not reach behaviour, Crafter's produces two achievements the matched control never performs.

Each rung, achievement by achievement. Running the same comparison along the whole ladder—with each arm identified by reproducing its rung size (+15.67, +4.08, +6.90 against the stated +15.7, +4.1, +6.9)—separates two kinds of rung. The two abstraction rungs both raise a tier the arm below barely reaches, though not identically: the capability bonus opens the iron tier from exactly zero (above), while the workshop bits lift the diamond from near-zero rather than from nothing ($0.9 \rightarrow 5.6\%$). Retention opens nothing—no achievement crosses from below 1% to above 5%—and instead converts those openings into reliability: iron pickaxe +27.6, collect iron +20.8, iron sword +17.6, diamond +16.3. Two of these four separate cleanly at the seed level—every retention seed exceeds every pre-retention seed for the iron pickaxe ($64.8/74.6/79.7$ against $38.3/47.7/50.4$) and for collecting iron—while two do not: one pre-retention seed reaches 59.4 on the iron sword against retention seeds at 57.8 and 59.0, and the diamond spans 7.4 to 33.2 under retention against 4.3 to 7.8 before it, so that gain rests on two of the three seeds. It also repairs much of the collateral cost the first rung incurred (skeleton +15.2, plant +8.3, sapling +8.2). Retention therefore acts on capabilities that counting has already made reachable rather than reaching new ones itself; this describes what retention does within the bonus arm and is not itself a test of the interaction reported in Table 3.

The cost the ladder incurs is structured. One cost persists throughout—the wood sword's mean falls at every rung ($98.7 \rightarrow 82.2 \rightarrow 80.2 \rightarrow 66.3$), though only the first and last steps separate at the seed level and the middle one lies within the seed spread. The cost the first rung itself incurs is solid per seed: for the wood sword, skeleton, stone sword and zombie alike, every bonus seed sits below every control seed. Nor is the cost diffuse: within each tier the tool that gates the technology tree is preserved while its non-gating sibling is not. The wood and stone pickaxes hold across the whole ladder ($99.9 \rightarrow 99.9$ and $97.4 \rightarrow 97.8$) while the wood and stone swords fall to 66.3 and 73.8, and the two combat achievements that depend on swords dip with them and partly recover under retention (skeleton $89.8 \rightarrow 73.8 \rightarrow 67.4 \rightarrow 82.7$). A natural reading is that the bonus reallocates a shared wood and stone budget toward the tools that open the next tier at the expense of those that do not; we report the pattern and do not measure allocation directly. The trade does track what the bonus opens rather than the mere presence of the bonus: in a configuration where the bonus is worth almost nothing (the minimal configuration of Table 4, $40.0 \rightarrow 40.5$) no tier opens—the iron pickaxe stays at

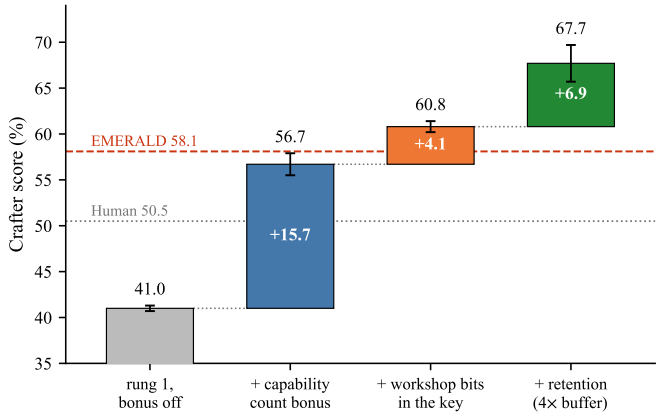


Figure 5: The decomposition ladder. Each rung differs from its neighbor by exactly one configuration flag (n=3 per rung; whiskers are seed sd): the capability-count bonus over the inventory-only key ($+15.7$), the workshop bits in the counted key ($+4.1$), and the $4\times$ replay buffer ($+6.9$).

≈ 0—and the swords move by only -5.1 each with overlapping seed ranges, against the clean -16.5 and -15.0 where the same bonus opens the iron tier.

Kitchen shows no comparable structure. Running the same per-element comparison on Kitchen finds no appliance consistently preserved or sacrificed: across the five matched pairs none favours one arm in all five, against an expectation of 0.38 such extremes under pure noise given six informative appliances at n=5. The per-seed differences change sign freely—the bottom burner runs $-73/+70/+89/-71/-26$ across the five pairs. Kitchen has no technology tree for a trade to be organised around, no appliance gating another, and correspondingly the enlarged configuration set has nothing to convert into. The boundary the score comparison reports is visible here in the per-element data. The observe-only arm passes the same test against the plain no-bonus arm—of five callable appliances none favours either arm consistently, and the microwave is identical in three of the five pairs—so maintaining the count table without ever rewarding it leaves per-element behaviour unchanged, as the control it serves as requires.

Crafter coverage trajectory. Against the matched observe-only arm, the bonus arm's configuration count is 2.4–$3.5\times$ at the full 10M budget (3171 vs. 895 configurations as logged; the observe-only count is logged per run segment, so its cumulative value lies in 895–1348 on average), n=5, with every matched pair favouring the bonus arm even at the observe arm's upper bound; the same holds at 5M (1.9–$2.7\times$), whereas the 1M and 2M windows ($1.7\times$ and $1.5\times$ as logged) are not resolvable under that bound and we draw nothing from them. The aggregate bonus does not decay over training in this agent: with the workshop bits the counted key reaches 2.7–3.3×10^3 configurations by 10M and is still growing, so the mean intrinsic reward stays at 0.015–0.020 from 1M to 10M on rungs 2–3 (rung 1 anneals to 0.004 by 8M and rises again as the iron tier opens new configurations); per-configuration annealing shifts exploration to the frontier rather than switching the bonus off, and the same holds on Franka Kitchen, where the bonus arm keeps discovering configurations to 3M. At the endpoint the arms separate completely—the lowest bonus seed reaches 2889 configurations against the highest observe-only seed's 963 as logged and 1608 at its upper bound. The sequence is not monotone, so we report the four windows and draw no conclusion from their ordering. **6M reference:** CFE+retention @6M $= 61.2 \pm 2.2$.

A.1 Learned capability representations: the full experimental report

The framework's full form: learned capability representations. Under P1–P3 the textbook bonus becomes a curriculum over the technology tree without any knowledge of the tree—and nothing in P1–P3 requires reading c from exposed state. The framework's full form *learns* the capability representation: a small set of discrete bits over the world model's latent state (P1), made monotone within a life by an explicit ratchet—the inductive bias that distinguishes an acquired capability from a visited place (P3)—and grounded in capability-change events rather than labels (P2). We have carried

Table 6: Task-structure ingredients used by deep-frontier Crafter methods, and whether each collects a diamond in a non-trivial fraction of episodes. Among structured methods only SEA reaches the diamond, and only with a recovered graph, an offline data-gathering phase driven by a separately trained IMPALA policy, and $\sim$50$\times$ compute (500M steps) on a modified environment in which the agent cannot lose health and 21 achievements are counted. On that variant SEA also reports a higher aggregate score (75.52) than any agent in Table 1; that number is computed over its own 21-achievement set and is not on the official 22-achievement scale used throughout this paper. EMERALD invests in a heavier world-model architecture and grazes the floor (0.5%). "n/r" marks a method whose Crafter evaluation does not report this quantity: ELLM is assessed on pretraining coverage and downstream tasks rather than on achievement rates. Two column conventions matter for reading the table fairly. "Intrinsic reward" marks any non-environmental reward signal, so ELLM counts: its signal is goal-based (rewarding LM-suggested goals) rather than novelty-based, and the novelty space is the axis this paper varies. "Heavy arch." marks a method whose route to the frontier is architectural investment; it is *not* a parameter count, and on that measure our agent is the larger of the two ($\sim$50M vs. EMERALD's 30M; Table 1). ‡CFE's base agent inherits an auxiliary achievement-prediction head and a 22-dim achievement encoder input from its published recipe, both ablated in Table 4 (the head is not load-bearing; the encoder input is worth 9.7 points at rung 1). The distinction we draw is narrower than "no achievement information": no achievement information enters the *exploration bonus*, which counts generic inventory state. EMERALD, the strongest comparator, uses neither ingredient—it trains on observations and reward alone—and still reaches 58.1, above our rung-1 bonus-off score of 41.0.

Method	Achv. graph	Achv.-keyed training	LM goal prior	Heavy arch.	Intrinsic reward	Crosses diamond
SEA (Zhou & Garg, 2023)	✓	✓	×	×	×	✓
Achv. Distill. (Moon et al., 2023)	×	✓	×	×	×	×
ELLM (Du et al., 2023)	×	×	✓	×	✓	n/r
EMERALD (Burchi & Timofte, 2025)	×	×	×	✓	×	×
CFE (ours)	×	✓‡	×	×	✓	✓

Table 7: Same mechanism, different novelty space (identical agent, budget, and configuration; rung-2 stack, buffer 5×10^5). Only the exploration signal differs.

Exploration signal	Score (%)	Notes
Capability-configuration count (CFE)	60.8 ± 0.6 (n=3)	iron pickaxe $\sim$ 45%
RND, real-reward pathway	43.4 ± 4.0 (n=5)	learned novelty
No bonus (as in Table 3)	42.7 (n=2)	diamond 0%
SimHash count, latent features	41.8 ± 1.0 (n=5)	hashed-state count
same, counted at collection time	39.9 ± 1.8 (n=3)	counting schedule matched to CFE
SimHash count, pixels	43.3 ± 4.5 (n=5)	hashed-state count

this program through a first round of controlled online experiments. The representation itself is learnable: offline, on replay data containing deep-tier behavior, probes recover the full 22-achievement tree from the learned bits (per-achievement AUC 0.94–0.95). Using the learned code as the *counted key* online, however, no variant yet clears the no-bonus floor—and each shortfall is attributable to a specific, correctable cause. The experiments isolate three conditions a functional learned key must meet; we present them as necessary, not yet shown sufficient. (i) *The key must be piecewise-stationary.* With the counting head allowed to drift, the key fragments ($>10^5$ configurations against the state-read key's $\sim10^2$–10^3) and the bonus never anneals, taxing the policy below every bonus-off score we report (32.2 ± 2.7 vs. 40.0–42.7; Tables 3 and 4); freezing the head between scheduled snapshots removes the fragmentation (configurations $\sim10^3$–10^4, the bonus falls to a third of the fragmented level; 34.9 ± 1.8). (ii) *Sparsity belongs on transitions, not occupancy.* An occupancy prior on bit firing rates structurally excludes persistent capabilities—holding an iron pickaxe is true for over a third of steps on a competent policy—and bits trained under it probe at chance (best capability AUC 0.50); moving the same budget to the flip rate, matching P3's once-per-life semantics, lifts the same probe to 0.75. (iii) *The count table and the policy's learned routines are coupled.* Resetting the count table under a competent policy re-inflates the bonus everywhere, and the induced retargeting of

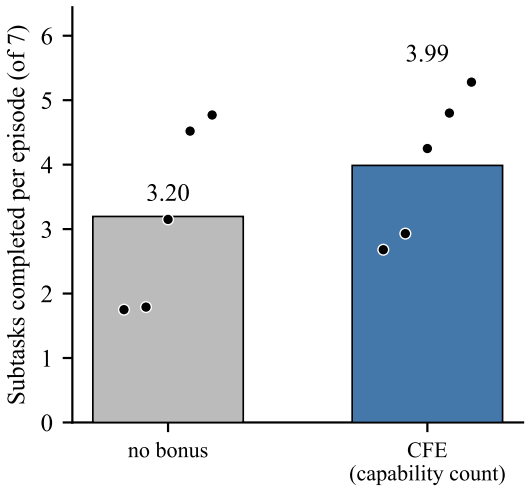


Figure 6: Franka Kitchen at 1M steps (dots are seeds, n=5 per arm; metric is the mean number of seven appliance subtasks completed per episode, over 100 evaluation episodes). The bonus arm averages 3.99 ± 1.14 against 3.20 ± 1.44 without it, a gap that an unpaired permutation test does not distinguish from chance ($p = 0.33$; likewise $+0.59$, $p = 0.55$, at 3M with n=6), although at 1M all five matched pairs favour the bonus arm (sign-flip $p = 0.0625$). The informative comparison on this domain is not the score but the coverage: against a matched arm that counts capability configurations without ever adding the bonus to reward, the bonus arm ends training having reached about twice as many (52 vs. 23, all five pairs) at indistinguishable score.

the reward predictor degrades the fragile deepest-tier skills before the table anneals (48.5 with the bonus switched on against an empty table vs. 53.7 with the bonus off, both warm-started from the same checkpoint, n=1 each; the bonus-on, table-carried-over control was not run)—consistent with the breakthrough-forgetting reading of Table 3. Full-budget runs with all three requirements met close the loop. With the counting head snapshot-frozen and the transition-sparsity prior in place, a purely learned count key reaches 37.8 ± 2.1 (n=3, 10M)—above the fragmented and occupancy-prior variants (32.2, 34.9) but still at the no-bonus floor, and a hybrid key that appends the explicit inventory bits to the learned bits lands identically (37.8 ± 3.7), well below the 56.7 the explicit bits earn on their own. On both, the deepest tier stays closed (iron pickaxe $\approx 0\%$). The learned code is thus *representable* (offline probe AUC 0.94–0.95), *online-learnable* under the transition prior (probe AUC 0.75), and *online-stable* (fragmentation removed), yet not *usable as a count key*: even co-located with the explicit bits it acts as a net tax on the policy rather than a curriculum. Two effects, each isolated above, remain: a learned code cannot lead the policy to a frontier it has not yet reached (the bonus never fires where the code has never advanced), and counting over noisy learned bits perturbs the reward predictor the policy relies on. The second effect is not a simple function of bonus size, nor of how many learned bits the key carries. Scaling the hybrid key's bonus to $0.3\times$ leaves it at the same level (36.0 ± 3.0, n=2); halving the code from twelve bits to eight makes it worse rather than better (33.1 ± 0.3, n=2); and at four bits training diverges outright on all three attempts, before one million steps. Across the range in which the learned code trains at all, then, the interference shrinks with neither the magnitude of the injected reward nor the size of the learned key—it rides the fact that the counted key is learned. Making the representation learnable is therefore necessary but not sufficient; a usable learned key needs a bonus coupling that does not feed representation noise back into the policy—the concrete open problem this analysis isolates. The component analysis (Table 4) identifies exactly which state-read channels such a representation must replace, and the ladder sets the bar it must clear—$+15.7$ is what the state-read representation is worth.

Formulation. A linear head on the world model's latent feature $\phi(s_t)$, which is detached so that bit learning never alters the world model, gives K bit probabilities and hard bits with a straight-through gradient; within a life the bits are ratcheted:

$$p_t = \sigma\big(W\,\mathrm{sg}[\phi(s_t)]\big), \qquad b_t = \mathbb{1}[p_t > \tfrac{1}{2}]\ \text{(straight-through)}, \qquad c_t = \max(c_{t-1}, b_t), \quad c_0 = 0.$$

Four losses train the head, none of which sees an achievement label. Let $e_t \in \{0,1\}^{16}$ mark the inventory slots that increase at step t and $f_t = \max_{1 \le i \le H} e_{t+i}$ (within the episode, H=8). (i) Sufficiency: $\mathcal{L}_{\text{suff}} = \text{BCE}\big(g(c_t), f_t\big)$ for a small predictor g, so the bits must carry what predicts the next acquisition. (ii) Monotonicity: $\mathcal{L}_{\text{mono}} = \sum_t \text{relu}(p_{t-1} - p_t)^2$. (iii) Transition sparsity: with $\bar{f}_k$ the mean rate of $0 \to 1$ flips of bit k (soft, on p), $\mathcal{L}_{\text{flip}} = \sum_k \big[\text{relu}(10^{-3} - \bar{f}_k) + \text{relu}(\bar{f}_k - 0.02)\big]$, a band that permits persistent bits but forbids chatter (the "occupancy prior" variant instead penalises the mean activation ρ_k towards 0.05). (iv) Controllability: a symmetric InfoNCE (temperature 0.1) between an embedding of the action chunk $a_{t:t+4}$ and an embedding of the flips $\text{relu}(b_{t+1} - b_t)$. The count key is the ratcheted output of an exponential-moving-average copy of the head (τ=0.005) with per-bit thresholds; "snapshot-frozen" means the counting head and thresholds are copied every 10^6 steps and held fixed in between, the count table being halved at each copy. The bonus over the learned key is the count term of Equation 1 with two small frontier terms, $r_t^{\text{int}} = \text{clamp}\big(b_N(c_t) + 0.25\,\text{relu}(b_N(c_t) - 0.5\,b_N(c_{t-1})) + 0.25\,\sum_k w_k\,\text{relu}(c_{t,k} - c_{t-1,k}),\ 0,\ 0.5\big)$ with $b_N(c) = 1/\sqrt{N(c)+1}$ and per-bit weights w_k that track how often bit k's flips are predictable from actions; K=12 in the runs reported above unless stated.

A.2 Replication on a second agent (EMERALD)

A 2×2 of {22-bit achievement input} $\times$ {capability bonus} on the EMERALD agent at 10M steps (official 256-episode evaluation); the primary quantity is the interaction $(E3 - E2) - (E1 - E0)$ on clean, never-resumed seeds, and resumed seeds are excluded from every reported value. Table 5 reports the four cells; the interaction is $+5.3$ (SE 6.9; 95% CI $[-8.5, +19.0]$; within-arm sd 7.6 on 16 df; minimum detectable effect at t=2: 13.8). The interval covers zero and the Crafter-sized $+15.7$ alike, so the replication is inconclusive at this precision: it neither reproduces the conditionality nor rules it out. The pixels-only, no-bonus cell is a fresh set of five seeds of the same configuration as the reproduction quoted above (46.9 ± 8.5); the two agree within their spreads.

How the achievement input enters EMERALD. At the encoder only: the 22-bit vector is broadcast to 22 constant 64×64 planes and concatenated to the RGB image as extra input channels of EMERALD's convolutional encoder; the decoder, the spatial latent, and the MaskGIT predictor are unchanged and reconstruct RGB alone. This is the simplest way to give a pixel agent the same information, not an architecture-matched one; the drop it causes ($52.9 \to 42.0$) may be a mismatch between a flat input and a spatial latent, which the interaction cannot separate from the scaffold effect.

A.3 Milestone persistence across arms

Table 8 reports, for every arm, when the first periodic in-training evaluation with an iron pickaxe occurs and in how many of the $\sim$1000 evaluations the iron pickaxe and the diamond are present; official 256-episode rates are repeated for reference. Every arm, the no-bonus floor included, reaches the iron tier at least once; the arms differ in how often it is held, and retention acts on the diamond.

B Additional reduced-configuration results

In a jointly-reduced configuration (no ϵ-prior, no workshop bits, $4\times$ buffer; achievement channels kept; n=3 per arm) the bonus does not separate from its controls: bonus on 42.0, bonus off 45.9, RND 40.9, SimHash-features 40.8, SimHash-pixels 40.1, base 39.7. This multi-flag configuration motivates the single-flag component removals of Table 4, and those removals show the effect is attributable to no single component: removing either the achievement observation or the crafting-action ϵ-prior leaves the bonus at 47.0 and 47.5 against matched floors of 40.5 and 42.8 (the same removals with the bonus off; Table 4), i.e. mean gains of $+6.5$ and $+4.7$ points on three seeds, with the bonus-on cells varying by 5.7 and 7.2. The deep-tier gain survives in some seeds and not others, and only the minimal configuration with all auxiliary components removed collapses to its own bonus-off score (40.5 against 40.0). The capability-frontier gain thus rests, in our agent, on the capability-aware scaffold as a whole rather than on any one privileged input—which is why the framework's learned form (Section 8) must replace the scaffold collectively, not a single channel.

Table 8: Milestone persistence (medians over seeds). Every seed of every arm crafts an iron pickaxe in at least one of its ∼1000 periodic in-training evaluations; the columns give the first such evaluation (M steps) and the number of evaluations in which the iron pickaxe and the diamond are present (an achievement counts as present when its time-weighted unlock count is positive), with official 256-episode rates for reference.

Arm	n	**first (M)**	**evals iron**	**evals diam.**	**iron %**	**diam. %**
no bonus (floor), buffer 5×10^5	2	2.9	35	1	2.3	0.0
no bonus, buffer 2×10^6	2	3.6	42	0	2.7	0.0
SimHash count, pixels	5	5.1	9	0	0.6	0.1
SimHash count, latent features	5	4.1	5	0	0.2	0.0
SimHash count, features, coll.-time	3	4.5	3	0	0.3	0.0
RND, real-reward pathway	5	2.6	15	0	11.2	0.2
rung 1: capability count, inventory key	3	4.0	114	5	41.0	0.9
rung 2: CFE default	3	2.3	536	107	45.4	5.6
rung 3: CFE + retention	5	2.2	578	292	72.5	20.5

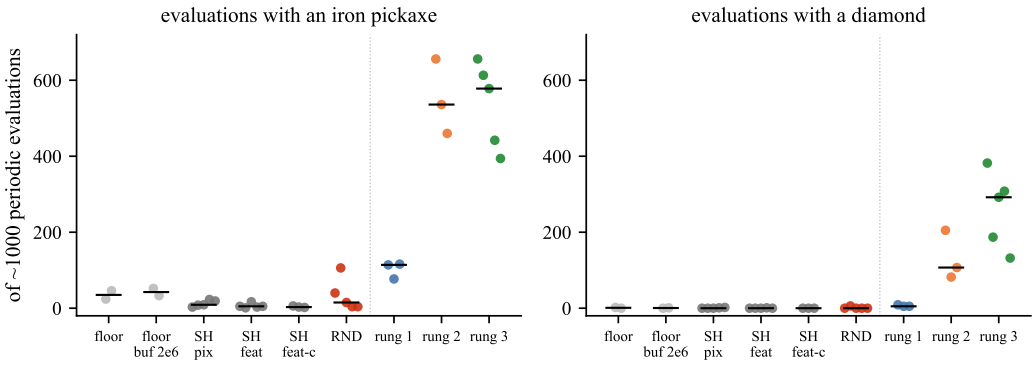


Figure 7: Milestone persistence per run (dots are seeds; bars are medians): the number of the ∼1000 periodic in-training evaluations in which an iron pickaxe (left) and a diamond (right) are present. Every run of every arm reaches the iron tier at least once; the arms differ in how often it is held, and retention acts on the diamond.

C BOUNDARY DOMAINS: FULL RESULTS

This appendix gives the full statistics behind the boundary-domain summary of Section 7; the paragraphs are the complete versions of the ones summarized there.

Meta-World pick-place: per-seed values and tests. c has ten bits: the grasp and success indicators, a one-hot band of gripper-to-object distance with edges at 0.05, 0.10, 0.20 m, and a one-hot band of object-to-target distance with edges at 0.07, 0.15, 0.30 m. Sparse reward, evaluation success as the mean of the final ten in-training evaluations (100 episodes each): bonus arm 1.00/0.14/0.99/1.00/1.00 (n=5, mean 0.83); matched sensors-only arm 1.00/0.05/0.00/0.00/0.04 (n=5, mean 0.22); base agent 0.00 (n=3). Exact permutation test on arm means $p = 0.048$; on the 4/5 vs. 1/5 success split, Fisher $p = 0.21$. Faithfulness of the sparse task was checked against a separate codebase (DrQ-v2 (Yarats et al., 2022), final gate mean ≈0.6), which lands inside the published band for this task.

It points the same way on sparse-reward Franka Kitchen. The Franka Kitchen task (Gupta et al., 2019) has no technology tree; its capability configuration is the set of completed appliance subtasks (a seven-bit vector over the microwave, kettle, two burners, light switch, and two cabinets—the manipulation analog of inventory), and the score is how many of the seven an episode completes. We ran both arms at two budgets, scoring every run over 100 evaluation episodes (Figure 6). The bonus arm is ahead on average at both: 3.99 ± 1.14 vs. 3.20 ± 1.44 at 1M steps (n=5, +0.79) and 4.71 ± 1.00 vs. 4.12 ± 1.73 at 3M (n=6, +0.59). Neither difference is distinguishable from

chance under an unpaired permutation test ($p = 0.33$ and $p = 0.55$), though at 1M the bonus arm is ahead in all five matched pairs (sign-flip $p = 0.0625$, the minimum five pairs can produce), and no arm completes the hinge cabinet in evaluation, so the attainable maximum here is six rather than seven—a bound one no-bonus seed actually reaches. The seed-to-seed spread dominates: the observed effects ($d = 0.61$ and $d = 0.41$) would need roughly forty and ninety seeds per arm to resolve at conventional power. We therefore report the score comparison as directionally consistent but unresolved, and do not treat it as evidence for the mechanism.

To ask whether the bonus does what it is supposed to do irrespective of the score, we re-ran the bonus-off arm with the count table still maintained but never added to the reward, so that the bonus is the only manipulated variable across matched seeds. By the end of training the bonus arm has visited about twice as many capability configurations (52 vs. 23; all five matched pairs favour it, sign-flip $p = 0.0625$, the smallest value five pairs can produce), while the two arms' task scores are indistinguishable (3.99 ± 1.14 vs. 3.67 ± 1.05, $p = 0.56$). The coverage difference is therefore not a by-product of one arm simply playing better. It is not yet distinguishable at 200k steps ($1.34\times$, $p = 0.56$), so it accumulates over training rather than being present from the start. On this domain the mechanism does what it claims—it enlarges the set of capability configurations the agent reaches—and that enlargement does not convert into task score.

Craftax. On Craftax the capability configuration is the fourteen inventory items thresholded at *holds at least one* (wood, stone, coal, iron, diamond, sapling, pickaxe, sword, bow, arrows, torches, ruby, sapphire, books) together with the dungeon-floor index, counted exactly (2^{18} cells, no hashing). At the 1M budget its authors propose as a challenge setting, the bonus ported into the official PPO baseline on Craftax-Symbolic-v1 is null (n=5, $+0.02$ percentage points). The benchmark's own study reports the same outcome for the standard toolkit—none of RND, ICM, or E3B improves on plain PPO there, and E3B hurts—and attributes it to a reward structure already dense enough to learn from without an intrinsic term (Matthews et al., 2024): their diagnosis and our prediction name the same cause. Extending our own agent (on Craftax-Pixels-v1) to ten times that budget does not change the verdict: at 10M steps, on the benchmark's 100-episode evaluation protocol, the bonus arm's normalised score is 7.65 (7.73/7.57) against the bonus-off arm's 7.62 (7.32/7.92), two seeds per arm, both reaching the same 22 achievements. The gap is not resolvable, and we do not read it as a direction: the two bonus-off seeds differ from each other by more than the arms differ, single-window values span 12.1–19.6, and narrowing the window to three evaluations reverses the sign. The count bonus has also fully annealed by then (257 configurations visited in 10M steps)—the difficulty-weighted reward leaves no under-priced frontier for it to find.