TABULA · embodied discovery

Jiawen Sun

1 · PROBLEM

what and why

Open-ended embodied agents are evaluated almost entirely in Minecraft. Voyager and its successors report autonomous discovery — novel items found, tech-tree depth reached, diamonds mined — as evidence that an LLM agent placed in a world can infer its rules and bootstrap its own skills. But Minecraft's rules are among the most thoroughly documented artifacts on the internet. Wikis, tutorials, and millions of transcribed playthroughs sit in every frontier model's pretraining corpus. The agent knows the crafting graph before it takes a single action. Every published discovery metric is therefore confounded: none of them separate rule inference from recall. The field's central empirical claim has never been tested against its most obvious alternative explanation.

2 · APPROACH

model / method, and why this one

TABULA is a rule-randomization layer over an open-world sandbox. Engine, action space, observation space, and task structure are held fixed; only the semantics are permuted. If an agent's competence is inference, it survives permutation. If it is recall, it collapses. TIER WHAT IS RANDOMIZED WHAT IT ISOLATES T0 · Vanilla Nothing — reproduces published setups Control; calibrates against prior results T1 · Recipe Crafting graph permuted; same items, new dependencies Memorized procedure T2 · Affordance Tool–material effectiveness and hardness tables permuted Memorized physical intuition T3 · Relabel T1+T2 plus item names replaced with novel tokens Residual lexical prior T3c · Control Novel tokens, vanilla rules Cost of unfamiliar naming alone T3c is the load-bearing control. Without it, a collapse at T3 could be explained by tokenization difficulty rather than by loss of world knowledge; with it, the two are separable. competence(T0) − competence(T3) → the recall gap competence(T3) − competence(T3c) → inference under novelty competence across successive T3 worlds → meta-discovery Rule-inference probe. Behaviour alone is ambiguous: an agent can stumble onto a correct craft by brute search without having inferred anything. So at fixed intervals the agent is asked to state, in text, the recipe or affordance it believes holds. Stated beliefs are scored against ground truth. This separates acting correctly from knowing why. Why a sandbox rather than a synthetic MDP. The claim under test is specifically about open-ended embodied agents, so the setting must retain what makes them interesting: partial observability, long horizons, compositional crafting, and an unbounded action space. A stripped-down MDP would test something else and prove nothing about Voyager-class results. Substrate. Primary experiments run in a Crafter-class 2D sandbox — fast, CPU-only, procedurally generated, with an explicit achievement tree that is trivial to permute programmatically. A reduced Minecraft replication via MineDojo validates that findings are not an artifact of the lightweight substrate.

3 · DATASET

source, size, preprocessing

There is no training corpus; the artifact is a distribution over environments. COMPONENT DETAIL Base substrate Crafter-class sandbox, open source, deterministic given seed Rule sampler Generates permuted crafting graphs, affordance tables, and token vocabularies from a seed Solvability Search-based planner that verifies every sampled world is fully completable before use, and records optimal depth-to-goal as a oracle normalizer World set 500 validated worlds per tier, held-out split of 100 never used during development Secondary MineDojo instance with a permuted recipe JSON, reduced task subset The solvability oracle is not optional infrastructure. Random permutation readily produces

worlds with unreachable goals, and an agent failing an impossible world is indistinguishable from an agent failing a hard one. Every world ships with a certificate of completability and a known optimal path length.

4 · METRICS

how you'll know it worked

MEASUREMENT METRIC INTERPRETATION Recall gap Tech-tree depth at T0 minus depth at T3 Primary result. A large gap means published discovery numbers are largely memory. Naming T0 minus T3c Subtracted from the recall gap; isolates unfamiliar-token cost penalty Rule inference Probe accuracy on stated recipes over time Distinguishes inference from successful search Sample Environment steps to first depth-k craft, normalized by Comparable across permuted worlds efficiency oracle optimum Meta- Slope of competence across successive unseen T3 worlds The positive result: does the agent get better at figuring out rules? discovery Significance Paired tests across matched seeds, 5+ seeds per cell Same world seeds across models and tiers Three frontier models plus one open-weights model, run under a Voyager-style scaffold held constant across all tiers so the scaffold is never a free variable.

5 · COMPUTE

training time, memory, dependencies

No model training. Cost is dominated by agent inference, which makes this cheap in hardware and expensive in tokens. PHASE WORK RESOURCE WALL CLOCK 1 Rule sampler + solvability oracle + world validation CPU only ~2 weeks 2 T0 / T3c calibration; reproduce a published baseline CPU + API ~4 days 3 Main sweep: 5 tiers × 4 models × 5 seeds CPU + API ~1 week 4 Meta-discovery: successive unseen worlds CPU + API ~5 days 5 MineDojo replication, reduced task subset 1 GPU + CPU ~1 week Token budget. Estimated 50–80M output tokens across the full sweep, assuming episode caps and aggressive context truncation in the scaffold. This is the binding constraint and the number most likely to be wrong; phase 2 exists partly to measure it before phases 3–5 are committed. Dependencies. Crafter-class sandbox · MineDojo + Malmo for the replication · a Voyager-style scaffold (skill library, iterative prompting, self-verification) reimplemented rather than forked, so it can be held identical across tiers · standard planning library for the oracle. No custom kernels, no distributed training.

6 · DELIVERABLES

model, paper, demo

1. TABULA generator. Open-source rule-randomization layer plus solvability oracle. Reusable by anyone evaluating embodied agents; the durable artifact regardless of what the results say. 2. Validated world sets. 500 certified worlds per tier with held-out split, versioned and released so results are reproducible. 3. Recall-gap table. Headline decomposition of published discovery metrics into recall and inference components, per model. 4. Meta-discovery curves. Competence across successive unseen worlds — the measurement that says whether anything general is being learned. 5. Reference scaffold. The held-constant Voyager-style harness, released so subsequent work can vary the agent rather than the plumbing. 6. Paper. Submitted regardless of sign. A confirmed collapse and a confirmed survival are both publishable, and both change how the next benchmark is built.

7 · RISK

what breaks first, and plan B

what breaks first, and plan B WHAT BREAKS PLAN B Agents fail T3 so completely that nothing is Graded permutation strength (permute k of n recipes) gives a dose–response curve instead of a floor. measurable This is built into phase 2, not bolted on later. T3c already collapses — novel tokens alone Then the design cannot separate naming from knowledge. Report that as a measurement finding and destroy performance fall back to T1/T2, which preserve familiar names. Random worlds are unsolvable or degenerate Solvability oracle gates every world; degenerate worlds (optimal depth below threshold) are rejected at sampling time. Result is unsurprising — everyone already Assumed is not measured. The

contribution is the size of the gap and the generator that lets others assumed recall dominates measure it. Token budget overruns Phase 2 measures true cost before phases 3–5 are committed; sweep is cut by model count, not by seeds. Someone publishes it first Live risk in a fast area. Mitigated by scope: the generator ships early and independently of the full sweep. KILL CRITERION If phase 2 shows that T3c performance is statistically indistinguishable from T3, the naming confound dominates and the headline experiment cannot be run as designed. Phases 3–5 are not committed; the generator is released with the negative methodological finding. — · PRIOR ART what this is not Voyager and its successors established the open-ended embodied agent paradigm; this proposal does not compete with them, it audits them. Procedural generation for generalization exists (ProcGen) but randomizes levels, not rules; XLand randomizes goals and games for agents trained from scratch, where contamination is not the concern. The distinct claim here is narrow and specific: for pretrained LLM agents, the benchmark world is inside the training corpus, and no published result controls for it. The nearest neighbours are the memorization-versus-generalization literature and recent work showing generative models can score well on surface diagnostics while holding incoherent underlying world models — this extends that critique from sequence models to embodied agents.