MapShift: Controlled Post-Intervention Evaluation for Embodied World Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Embodied agents are often evaluated in the same environment in which they were explored or trained, making it difficult to assess whether a learned world model supports planning after the world changes. Existing evaluations can conflate memory of the explored environment, belief update after change, and planning under the updated belief. We introduce MapShift, an executable benchmark for controlled post-intervention evaluation (CPE): an agent explores a base environment without a task reward; the environment is modified by a controlled intervention in the metric, topology, dynamics, or semantics; and the agent is evaluated on post-intervention planning, inference, and adaptation tasks. The contribution is measurement infrastructure: matched base/intervened pairs, family-wise estimands, severity ladders, invariant validators, benchmark-health gates, protocol-comparison tooling, and reproducible artifact generation. In the expanded 24-motif release, health gates pass with zero fatal leakage, zero task rejections, perfect reference solvability, no intervention-validator failures, and no severity-magnitude failures. A deterministic mechanism diagnostic shows that same-environment evaluation underestimates the belief-update advantage by 3x on topology shifts, from $\Delta = 0.304$ under CPE versus 0.102 same-environment, and by 7x on semantic shifts, from $\Delta = 0.724$ versus 0.102; planning slices also exhibit protocol-induced rank reversals.

1 Introduction

World models are useful only insofar as their learned structure remains actionable when the world does not stay fixed. In embodied evaluation, however, agents are often explored, trained, and evaluated under tightly coupled environment assumptions. Even procedurally generated benchmarks usually ask whether a system performs well under a sampled distribution, not whether it can use reward-free exploration of a specific world after a controlled change. This distinction matters because protocol design determines which competence is being measured: stale reuse of an explored map, explicit belief update after a change, or planning under the updated belief. Embodied planning combines partial observability and sequential decision-making with structured spatial state [Kaelbling et al., 1998, Sutton and Barto, 2018]; failures may arise from brittle geometry, missing connectivity, incorrect transition dynamics, or shifted cue meanings.

We approach this as an evaluation problem. Controlled post-intervention evaluation (CPE) samples a base environment e, lets an agent explore e without task reward for T_{exp}, applies an intervention $I_f(\sigma)$ to obtain e', and evaluates tasks from $Q(e, e')$. The family f declares what changed and the severity σ controls how strongly it changed. Motif and generator seed are held fixed while one environment factor changes, so the protocol tests post-intervention competence for a specific explored world. Same-environment evaluation tests whether a system can exploit a representation in the environment it observed; CPE tests whether that representation remains actionable after a declared change. Figure 1 summarizes the evaluation flow.

Submitted to the ICML 2026 RLxF Workshop. Do not distribute.

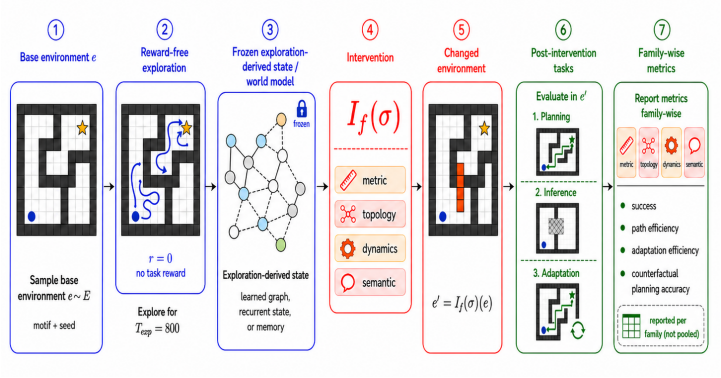

Figure 1: Controlled post-intervention evaluation protocol. An agent first explores a base environment without task reward, producing a fixed exploration-derived memory or world model. A controlled intervention then changes the environment along one family and severity level. The agent is evaluated only after the intervention on planning, inference, and adaptation tasks, and results are reported by intervention family to expose intervention-specific failure modes.

MapShift prioritizes auditability over visual realism. This is intentional: CPE requires knowing which factor changed, proving that non-target factors stayed fixed, verifying solvability, logging task rejection, and checking whether weak baselines saturate. These requirements are difficult to audit in photorealistic environments without first defining the measurement contract. MapShift is therefore a reference implementation of the contract, not a replacement for Habitat, ProcTHOR, or robotics benchmarks. The intended use is comparison of post-intervention robustness profiles and auditing whether world-model conclusions depend on evaluation protocol.

The paper is organized around one central evaluation-science question: when do protocol choices change conclusions about world-model competence? MapShift answers through family-wise reporting, protocol controls, and a deterministic mechanism diagnostic that separates stale maps, local heuristics, and explicit belief update.

Our contributions are: (i) a controlled post-intervention protocol for measuring embodied competence after reward-free exploration; (ii) an executable benchmark with 24 motifs, four intervention families, severities 0–3, planning/inference/adaptation tasks, $T_{\text{exp}} = 800$, matched-pair validators, health gates, and reproducible artifact generation; and (iii) reproducible diagnostics showing that CPE separates stale reuse from explicit update and exposes task-conditioned protocol rank sensitivity on held-out motifs.

2 Related work

Benchmarks for embodied agents. ALE, Gym, DeepMind Control/Lab, Procgen, and MiniGrid support reproducible agent research [Bellemare et al., 2013, Brockman et al., 2016, Tassa et al., 2018, Beattie et al., 2016, Cobbe et al., 2020, Chevalier-Boisvert et al., 2023]; Crafter and MineDojo add open-ended skills [Hafner, 2022, Fan et al., 2022]; and Habitat, AI2-THOR, ProcTHOR, VirtualHome, and ALFRED add richer embodied structure [Savva et al., 2019, Kolve et al., 2017, Deitke et al., 2022, Puig et al., 2018, Shridhar et al., 2020]. MapShift is complementary: it is a reference implementation for testing whether reward-free environment knowledge remains useful after a declared intervention. A ProcTHOR or Habitat port would need the same matched-pair invariants, rejection accounting, solvability checks, weak-baseline saturation checks, provenance checks, and protocol comparisons.

World models and reward-free evaluation. World-model and model-based RL methods learn predictive structure for planning and control [Ha and Schmidhuber, 2018, Hafner et al., 2020, Schrittwieser et al., 2020], while reward-free exploration separates information gathering from downstream reward optimization [Jin et al., 2020, Sekar et al., 2020, Mendonca et al., 2021]. WorldTest/AutumnBench also separates reward-free interaction from a scored derived test phase [Warrier et al., 2025]. The distinction is identifiability: MapShift declares the changed factor, constructs a matched pair, and runs protocol ablations on the same generated grid; the compact comparison below summarizes this difference.

Protocol distinction from reward-free derived tests.

Axis	Derived-test protocol	MapShift CPE protocol
Changed factor	Need not be a declared estimand	Declares metric, topology, dynamics, or semantic family/severity
Matched pair	May compare different derived tasks or worlds	Holds motif, seed, start, and non-target factors fixed across e and e'
Validity checks	Benchmark-specific documentation	Emits validators, leakage checks, rejection counts, solvability, and saturation gates

Interventions, shift, and documentation. Causal reinforcement learning studies confounding, interventions, and counterfactual reasoning in sequential decision processes [Lu et al., 2018, Buesing et al., 2019, Rezende et al., 2020]. MapShift uses known generator interventions for diagnostic robustness rather than estimating causal effects from observational trajectories. Distribution-shift and benchmark-ranking work shows that aggregate performance and rankings can hide robustness failures or depend on aggregation choices [Koh et al., 2021, Maier-Hein et al., 2018, Perlitz et al., 2024, Chen et al., 2025]. Documentation work argues that evaluation artifacts should state intended use, scope, limitations, and provenance [Gebru et al., 2021, Mitchell et al., 2019]; MapShift operationalizes this through claim boundaries, health gates, and reproducible generation commands.

3 Controlled post-intervention protocol

Protocol. Let $\mathcal{E}$ denote a distribution over embodied environments, $\mathcal{F}$ the set of intervention families, and Σ_f the severity ladder for family f. A base environment $e \sim \mathcal{E}$ is sampled and exposed to the agent for reward-free exploration. During exploration, the system M receives observations and actions but no downstream task reward. We write the resulting internal state as

$$z_M = \mathrm{Explore}(M, e, T_{\mathrm{exp}}), \tag{1}$$

where z_M may be a recurrent hidden state, an explicit memory, a learned graph, or an empty state in the no-exploration ablation. After exploration, an intervention operator $I_f(\sigma)$ is applied, where $f \in \mathcal{F} = \{\mathrm{metric}, \mathrm{topology}, \mathrm{dynamics}, \mathrm{semantic}\}$ and $\sigma \in \Sigma_f$ is a discrete severity level. The resulting environment is

$$e' = I_f(\sigma)(e). \tag{2}$$

A task $q \sim Q_c(e, e')$ is then sampled from task class $c \in \mathcal{C}$, where $\mathcal{C}$ contains planning, inference, and adaptation. The evaluated system is scored using z_M from exploration in e while solving or inferring in e':

$$S(M; e, f, \sigma, c, q) = \mathrm{Perf}(M, z_M, e', q)\,. \tag{3}$$

The population target of the benchmark is therefore

$$\mathrm{CPE}(M) = \mathbb{E}_{e\sim\mathcal{E},\, f\sim\mathcal{F},\, \sigma\sim\Sigma_f,\, c\sim\mathcal{C},\, q\sim Q_c(e,e')} \left[S(M; e, f, \sigma, c, q)\right]. \tag{4}$$

The main report is family-wise:

$$\mathrm{CPE}_f(M) = \mathbb{E}_{e,\, \sigma\in\Sigma_f,\, c,\, q} \left[S(M; e, f, \sigma, c, q)\right], \tag{5}$$

because the benchmark is designed to separate metric, topology, dynamics, and semantic failure modes.

Operational intervention semantics. The generator can be viewed as a deterministic map $G(u, x)$, where u contains motif identity, sample seed, and layout randomness, and $x = (x_{\mathrm{metric}}, x_{\mathrm{topology}}, x_{\mathrm{dynamics}}, x_{\mathrm{semantic}})$ contains the configurable environment factors. A family intervention constructs

$$e' = G(u, x_{-f}, \phi_f(x_f, \sigma))\,, \tag{6}$$

where x_{-f} denotes the non-target factors and ϕ_f is the family-specific transformation. This is the sense in which the benchmark creates matched post-intervention pairs: the background motif and seed are held fixed, while one declared factor is changed and validators check that non-target factors remain invariant. Each episode record stores the intervention provenance needed to audit the controlled test-time shift.

Empirical estimator. The executable study estimates $\mathrm{CPE}_f(M)$ with a finite set of motifs, seeds, severities, task samples, and model seeds. Let $\mathcal{D}_f$ be the set of valid episode records for family f and baseline M, after task rejection. The reported family-wise score is

$$\widehat{\mathrm{CPE}}_f(M) = \frac{1}{|\mathcal{D}_f|} \sum_{r \in \mathcal{D}_f} s_r, \tag{7}$$

where s_r is the task-normalized primary score for episode record r.

Primary score composition. For main family-primary tables, the primary score uses non-identity severities $\Sigma_f^\star = \{1, 2, 3\}$ and equal weights over severity and task class:

$$\widehat{\mathrm{CPE}}_f^\star(M) = \frac{1}{3} \sum_{\sigma \in \Sigma_f^\star} \frac{1}{3} \sum_{c \in \{\mathrm{plan}, \mathrm{infer}, \mathrm{adapt}\}} \frac{1}{|\mathcal{D}_{f\sigma c}|} \sum_{r \in \mathcal{D}_{f\sigma c}} s_r. \tag{8}$$

Records within each family/severity/task-class cell have equal weight. Severity 0 is included in release coverage, health gates, and severity-response sanity tables, but excluded from Table 2 and the main family-primary result tables unless explicitly labeled. The protocol delta in Table 2 is $\Delta_{\mathrm{protocol}}(f) = (\mathrm{BU-stale})_{CPE,f} - (\mathrm{BU-stale})_{\mathrm{same},f}$. The privileged planning reference upper-bounds planning feasibility, not the mixed family score.

Protocol comparison. Protocol sensitivity is measured by replacing pieces of the evaluation operator while holding the generated study grid fixed. Same-environment evaluation replaces e' with e at task execution time. No-exploration evaluation replaces z_M with an empty memory. Short- and long-horizon evaluations multiply task horizons by fixed factors. Rank changes are computed from the induced orderings

$$\pi_p(f) = \mathrm{rank}_M\left(\widehat{\mathrm{CPE}}_{f,p}(M)\right), \tag{9}$$

where p indexes the protocol variant. We summarize protocol sensitivity using Kendall τ, rank reversals, best-method changes, and family-wise rank changes.

4 Methodology

Executable study pipeline. The benchmark study is generated from versioned configuration files rather than manually curated episode lists. For each motif and sample seed, the generator writes a base environment, deterministic identifier, and manifest. Baselines run the configured reward-free exploration budget; learned methods train or load checkpoints keyed by baseline, environment identifier, model seed, and hyperparameter hash. Intervention operators and task samplers then produce the changed environments, task records, invariant checks, and rejection counts used by the health report.

Intervention construction. Table 1 summarizes the four intervention families. Each family changes one intended axis while preserving other axes when possible. Severity level 0 is the identity operator and is included as a sanity check; levels 1–3 increase the configured severity parameter. Severity is an ordinal within-family ladder; cross-family difficulty comparisons are handled empirically through family-wise reporting and health diagnostics. The chosen ladders span small, medium, and large changes for each family and are interpreted through health checks for reference solvability, weak-baseline non-saturation, intervention isolation, and within-family severity response.

Table 1: Intervention methodology. Each family has a scalar severity parameter, preservation constraints, and expected failure modes used for benchmark health checks and interpretation.

Family	Severity parameter	Values	Preserved quantities
Metric	action-gain scale	1.0, 1.1, 1.25, 1.5	topology, semantic identities
Topology	blocker fraction	0.0, 0.08, 0.18, 0.32	semantic identities, action response
Dynamics	friction multiplier	1.0, 0.9, 0.75, 0.6	geometry, semantic identities
Semantic	remap fraction	0.0, 0.1, 0.25, 0.5	geometry, action response

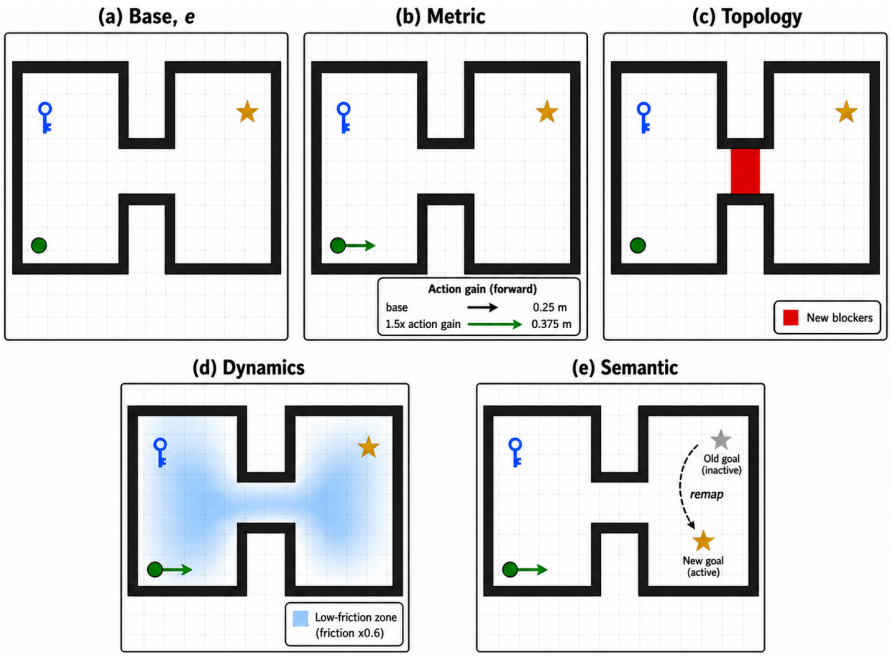

Figure 2: Matched intervention pairs in MapShift. A single base environment (a, two-room connector motif) is shown alongside the same environment after high-severity (level 3) interventions from each of the four families: metric (b, action-gain rescaling), topology (c, new corridor blockers), dynamics (d, low-friction zones), and semantic (e, goal-token remap). Background motif, seed, key location, and agent start are held fixed across panels; only the declared environment factor changes. Validators check that non-target factors remain invariant after each intervention. Severity-3 levels are shown for visibility; the full study uses severities 0, 1, 2, and 3 per family.

Environment and observation model. The reference substrate uses 96×96 occupancy maps at 0.25m resolution. The action interface is semi-continuous navigation with a 0.25m forward primitive and 15° turn primitive. Observations are egocentric and local: the agent observes a 3.5m radius, 100° field of view, and semantic channels when visible. Exploration uses a single episode, uniform navigable starts, and no privileged global state for non-reference baselines.

Exploration budget. The exploration budget is held fixed at $T_{\text{exp}} = 800$ steps for all motifs and non-reference baselines. A fixed budget makes exploration insufficiency part of the evaluated condition: a representation that works only after near-complete coverage should be distinguishable from one that remains useful under partial coverage. The benchmark health report therefore includes visited-cell ratios, visited-node ratios, path-length distributions, and weak-baseline saturation checks, so that under-exploration in larger motifs is visible rather than hidden by the protocol.

Figure 2 shows the matched-pair construction used by these interventions.

Worked example. In a topology CPE episode, the agent may explore a two-room connector map and store a traversable corridor. After exploration, blockers are inserted into that corridor while token identities and dynamics are preserved. A stale-map planner routes through the old corridor; a belief updater must use local post-shift observations to repair the map before planning. Same-environment evaluation would score the stale route as valid, while CPE scores it in the changed environment.

Intervention validation. After each intervention, validators check that the intended factor changed and declared invariants were preserved: metric shifts preserve topology and semantics, topology shifts preserve cue identities and action-response parameters, dynamics shifts preserve geometry, and semantic shifts preserve navigable geometry and reachability. These health checks are never model-facing; failures are reported before model results and block a valid release artifact.

Task sampling. The three task classes use shared start, goal, visibility, and horizon policies across comparable conditions. Planning covers shortest-path navigation, rerouting, changed dynamics, and

changed cues; inference covers topology-change detection, masked-region prediction, and counterfactual reachability; adaptation covers limited post-shift interaction followed by replanning. Planning horizons are 64, 96, and 128 steps; inference horizons are 16 and 32; adaptation budgets and horizons are 16, 32, and 64.

Task rejection and solvability. The sampler rejects tasks that cannot be solved by the privileged planning reference under the intended environment, tasks whose answer is unchanged in a way that makes the post-intervention query vacuous, and tasks whose shortest path or horizon makes the episode trivial. The artifact records rejection counts by reason and by benchmark cell. The health report therefore distinguishes a genuinely difficult benchmark cell from a cell that was under-covered because valid tasks could not be sampled.

Baseline evaluation. All non-reference baselines share $T_{\mathrm{exp}} = 800$. The privileged post-intervention planning reference provides a solvability/planning diagnostic with access to the changed environment. The same-environment reference is a protocol control: it asks what would be concluded if tasks were executed in the explored environment. The stale-map planner is a sensitivity floor that stores the pre-intervention map and never revises it after the intervention. Classical belief update stores occupancy/token bindings and updates from local post-shift observations; it receives no intervention label, privileged mismatch locations, or global post-intervention map. Learned baselines use fixed hyperparameters for pretrained graph, persistent memory, relational graph, and structured dynamics. Each record stores the baseline, run identifier, model seed, protocol, motif, split, family, severity, task class/type, and grouping key.

Weak heuristic floor. The weak heuristic is a meaningful but limited floor. It stores visited cells, visible token bindings, and shallow local traversability, then plans only over remembered local structure without global map repair. It intentionally fails under structural revision and stale semantic bindings, making it stronger than random guessing but too weak to saturate post-intervention reasoning. Full deterministic baseline behavior is specified in Appendix B.

Information access. Non-reference methods receive only their exploration trace, task specification, and internal memory; the global map and intervention label are reserved for explicit reference baselines.

Scoring. Planning episodes record success, observed path length, privileged-reference path length, normalized path efficiency, reference path gap, and post-intervention planning correctness. For a planning record r, path efficiency is

$$\eta_r = \begin{cases} \min(\ell_r^*/\ell_r, 1) & \text{if the agent succeeds and } \ell_r > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where ℓ_r^* is the shortest feasible reference path length in the evaluation environment and ℓ_r is the observed executed length. Post-intervention planning accuracy is the success rate on planning tasks whose correct answer depends on the intervention, excluding identity shortest-path sanity tasks. These planning-reference quantities are kept separate from the mixed family primary score: family scores aggregate planning, inference, and adaptation components and are not normalized to make the privileged planning reference an upper bound.

Inference episodes record predicted and expected answers and are scored by task type. Masked-region prediction uses exact-match accuracy for the changed family label or masked state category; post-intervention reachability uses exact-match accuracy for the changed semantic token's target node or room in e'. Topology-change detection uses AUROC for probabilistic outputs and binary confidence scores for deterministic baselines; the implementation computes average-rank AUROC with tied ranks. Adaptation episodes record a recovery curve over the allowed post-shift interaction budget and summarize it with adaptation sample efficiency. The family primary score is computed from task-relevant components and is used for ranking only after family-wise aggregation. Confidence intervals use 1000 bootstrap resamples at 95% confidence, grouped by environment/model-seed unit so that repeated task records from the same environment/model-seed combination are not treated as independent.

5 MapShift benchmark design

MapShift uses a partially observed navigation substrate with occupancy, transition, and semantic layers. The design exposes geometry, connectivity, dynamics, and semantics independently while remaining cheap enough for many seeds, protocol comparisons, and health checks. Its purpose is to make intervention and validation machinery inspectable as a reference artifact for future wrappers. The main release uses 24 motifs, four families, severities 0–3, three task classes, three samples per class/cell, $T_{\text{exp}} = 800$, and 1000 grouped-bootstrap resamples; full release values are listed in Appendix A. Splits are motif-level, with eight held-out test motifs for leakage-controlled protocol comparison.

6 Benchmark health and validity checks

An evaluation benchmark should be evaluated before its models are. The MapShift artifact therefore generates a benchmark health report before writing model results, covering motif splits, family/severity/task coverage, task rejection, reference solvability, weak-baseline non-saturation, path and horizon distributions, severity magnitude, intervention isolation, and leakage.

Task rejection is counted after built-in resampling, so zero task rejections means the released grid contains valid, non-vacuous tasks in every motif/family/severity/task-class cell, not that no internal proposal was discarded. The severity-magnitude gate checks configured intervention magnitude, not model-score monotonicity: scalar intervention magnitude must be nondecreasing from severity 0 to 3, while empirical scores may be nonmonotone because the primary score mixes tasks and finite samples. Saturation would occur if the privileged reference and weak heuristic were both near-perfect, or if severity curves were flat because all methods solved or failed every cell; MapShift treats such cases as health warnings.

Leakage is reported in three categories. Fatal leakage corresponds to split or task leakage that invalidates a result and must be zero. Diagnostic warnings identify suspicious patterns requiring inspection. Benign reusable templates are explicitly labeled with rationale, such as shared query forms that do not reveal held-out environments. In the completed run, fatal leakage, leakage warnings, intervention-validator failures, severity-magnitude failures, and final-release task rejections are all zero, while reference solvability is 1.000.

7 Experimental program

Baselines and role. The study reports protocol/reference baselines plus weak heuristic, stale-map planner, classical belief update, pretrained graph, persistent memory, relational graph, and structured dynamics. The learned baselines are calibration probes, not a state-of-the-art leaderboard: they test whether capacity, persistent memory, relational structure, or factorized dynamics alone substitute for explicit post-intervention update. The main mechanism claim uses deterministic/reference baselines because they isolate stale reuse, local heuristics, and belief update.

Study grid and aggregation. Trace-trained learned baselines use five model seeds; the larger pretrained graph and persistent-memory rows are reported on 2592 non-identity episodes. Hyperparameters are specified in versioned configs before final evaluation, validation motifs are reserved for pre-final choices, and bootstrap resampling is grouped by environment/model-seed unit. The mechanism diagnostic uses the same generated grid and compares CPE to same-environment and no-exploration variants, reporting rank statistics plus paired stale-map versus belief-update deltas.

Reviewer-facing artifact path. Commands are in the README; expected runtimes and outputs are shown.

Path	Expected output
CPU smoke test (2–5 min)	Small records, configs, and rendered sanity tables
Health audit (minutes after records)	Zero fatal leakage/rejections/validator failures; solvability 1.000
Table/Figure reproduction (minutes from records; 30–36 h full L4 rerun)	Matching Table 2, Figure 3, CIs, and assets
MiniGrid smoke (minutes)	24 envs, 384 pairs, zero failures, solvability 1.000

8 Empirical calibration findings

The reference studies calibrate the measurement infrastructure and protocol-sensitivity diagnostics rather than establishing a definitive leaderboard. On the expanded release, all health gates pass: zero task rejections, leakage, intervention-validator failures, and severity-magnitude failures, with reference solvability 1.000 and weak-heuristic health 0.415. The central empirical result is protocol sensitivity: same-environment evaluation underestimates the advantage of explicit belief update over stale-map reuse by 3x on topology shifts and 7x on semantic shifts.

This is an evaluation-methodology finding, not merely the observation that an updater can beat a non-updater. Across all 24 motifs, classical belief update outperforms stale-map reuse when stored structure or semantics become wrong: topology score 0.689 versus 0.395, and semantic score 0.810 versus 0.178. Same-environment evaluation masks most of this gap. Stale-map reuse remains competitive on dynamics shifts, matching the planning reference at 0.656, as expected when geometry is preserved.

Table 2: Held-out mechanism deltas. Entries are classical belief update minus stale-map family scores on the 8 held-out test motifs and non-identity severities 1–3; brackets are paired grouped-bootstrap 95% confidence intervals.

Family	CPE gap	Same-env. gap	Protocol delta
Topology	0.304 [0.189, 0.424]	0.102 [0.082, 0.125]	0.201 [0.095, 0.306]
Semantic	0.724 [0.692, 0.749]	0.102 [0.080, 0.120]	0.622 [0.611, 0.631]

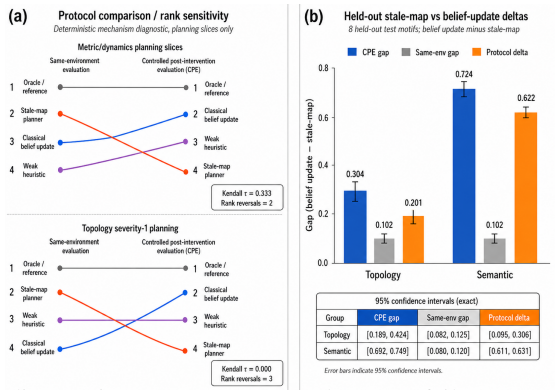

Figure 3: Mechanism diagnostic: CPE exposes stale reuse failures. Panel (a) compares same-environment and CPE conclusions; planning slices show rank reversals. Panel (b) reports held-out deltas: CPE gaps of 0.304 on topology and 0.724 on semantic shifts, versus same-environment gaps of 0.102.

Same-environment evaluation changes method order in declared task slices: metric/dynamics planning yields $\tau = 0.333$ with two rank reversals, and topology severity-1 planning reaches $\tau = 0.000$ with three. Aggregated mixed scores are deliberately conservative and combine planning, inference, and adaptation. At that level, rank order can remain stable while task-conditioned conclusions change. The protocol failure we target is therefore not that all aggregate rankings reverse, but that scientifically meaningful planning slices can reverse or collapse gaps under same-environment evaluation.

On the 8 held-out test motifs, belief update exceeds stale-map reuse under CPE in every topology and semantic motif; Table 2 reports paired grouped-bootstrap confidence intervals. Same-environment gaps shrink to 0.102 for both families, while protocol-reversal deltas remain positive.

To test whether CPE is a substrate-specific trick or a transferable measurement contract, we instantiate the same interface in a MiniGrid-compatible symbolic wrapper: a portability sanity check, not a pixel or language-conditioned evaluation.

Table 3: MiniGrid-compatible CPE sanity study in a symbolic MiniGrid-style wrapper.

Quantity	Result
Scale and health	24 environments; 384 matched pairs; 0 rejections/validator failures; reference solvability 1.000
Gaps and ranking	Belief-update minus stale-map CPE gap: 0.313 topology, 0.750 semantic; semantic same-env vs. CPE $\tau = 0.667$ with one reversal
Scope	Symbolic wrapper and portability sanity check; not a pixel/VLA or robotics-realism result

Table 4 summarizes how these checks change the interpretation of common benchmark conclusions. The point is not that every aggregate leaderboard changes, but that specific conclusions about stored world models, update mechanisms, and post-change planning require a protocol that declares the intervention and audits the matched pair.

Table 4: Conclusion audit enabled by MapShift. Standard or underspecified protocols invite the left-hand conclusion; MapShift supports the middle diagnosis. The source column separates the full learned-probe run from the focused mechanism diagnostic.

Standard conclusion	Source	MapShift diagnosis	Why it matters
Same-env task ranking is enough.	24-motif mechanism diagnostic	Planning slices show protocol-induced reversals: metric/dynamics have $\tau = 0.333$ with two reversals; topology severity-1 planning has $\tau = 0.000$ with three.	Standard protocols can change method order.
Stale maps are competitive.	Mechanism diagnostic	Belief update beats stale-map reuse on all 8 held-out topology and semantic motifs under CPE.	Updating, not reuse, is the target competence.
Memory capacity is enough.	Capacity runs	Large persistent memory improves to 0.313–0.354 but remains below update-based references on most families.	Capacity does not substitute for explicit update.
Same-env gaps estimate the intervention effect.	Paired deltas	CPE increases the belief-update advantage over stale-map by 0.201 on topology and 0.622 on semantic shifts.	Protocol assumptions change conclusions.

9 Limitations and future work

MapShift is not a deployment-realism benchmark and should not be interpreted as evidence of real-world robustness, language grounding, commonsense reasoning, or deployment readiness. **Substrate:** the release uses symbolic/semi-continuous navigation rather than 3D photorealism, contact dynamics, manipulation, or natural-language instruction following. **Baselines:** learned rows are calibration probes, not SOTA embodied agents. **Diagnostic scope:** deterministic mechanisms isolate stale reuse and update but do not prove that all learned agents will separate in the same way. **Metrics:** mixed family-primary scores support ranking within MapShift, but task-specific metrics should be inspected whenever the scientific claim concerns planning, inference, or adaptation in isolation.

Future work should carry the same CPE contract into richer substrates without relaxing the audit requirements. ProcTHOR, Habitat, robotics, and language-conditioned ports need declared intervention families, matched-pair validators, solvability and rejection accounting, weak-baseline saturation tests, same-grid protocol ablations, and checks that instructions do not leak interventions or target bindings. Longer-horizon and adaptive settings should log post-change observation timing so gains reflect world-model update rather than hidden oracle access.

Stronger-agent studies should evaluate pretrained embodied world models, active post-intervention information gathering, and map-revision policies under the same estimand and health checks; the current learned baselines are only modest calibration probes.

10 Conclusion

MapShift frames post-intervention embodied world-model evaluation as a measurement problem: it separates memory, belief update, and planning with matched reward-free exploration and controlled interventions. The artifact generates matched pairs, family-wise estimands, validators, health checks, protocol comparisons, and reviewer-facing records. The expanded diagnostic shows that same-environment evaluation can miss stale-reuse failures while CPE exposes held-out update gains on topology and semantic shifts.

More broadly, benchmark conclusions should be tied to an identifying protocol: declare the change, verify non-target invariants, and compare against same-environment and no-exploration controls. MapShift supplies that auditable contract for future richer embodied benchmarks.

References

Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Victor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. DeepMind Lab. *arXiv preprint arXiv:1612.03801*, 2016.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Lars Buesing, Theophane Weber, Yori Zwols, Nicolas Heess, Sébastien Racanière, Arthur Guez, and Jean-Baptiste Lespiau. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019.

Jianlyu Chen, Nan Wang, Chaofan Li, Bo Wang, Shitao Xiao, Han Xiao, Hao Liao, Defu Lian, and Zheng Liu. AIR-Bench: Automated heterogeneous information retrieval benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19991–20022, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.982.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *arXiv preprint arXiv:2306.13831*, 2023.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2048–2056, 2020.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. In *Advances in Neural Information Processing Systems*, 2022.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems*, 2022.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64 (12):86–92, 2021.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pages 2451–2463, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4870–4879, 2020.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Ben Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664, 2021.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3d environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Deconfounding reinforcement learning in observational settings. *arXiv preprint arXiv:1812.10576*, 2018.

Lena Maier-Hein, Matthias Eisenmann, Annika Reinke, Sinan Onogur, Marko Stankovic, Pascal Scholz, Tal Arbel, Hrvoje Bogunović, Andrew P. Bradley, Aaron Carass, et al. Why rankings of biomedical image analysis competitions should be interpreted with care. *Nature Communications*, 9(1):5217, 2018.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *arXiv preprint arXiv:2110.09514*, 2021.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019.

Yotam Perlitz, Ariel Gera, Ofir Arviv, Asaf Yehudai, Elron Bandel, Eyal Shnarch, Michal Shmueli-Scheuer, and Leshem Choshen. Do these LLM benchmarks agree? fixing benchmark evaluation with BenchBench. *arXiv preprint arXiv:2407.13696*, 2024.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. VirtualHome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018.

Danilo Jimenez Rezende, Ivo Danihelka, George Papamakarios, Nan Rosemary Ke, Ray Jiang, Theophane Weber, Karol Gregor, Hamza Merzic, Fabio Viola, Jane Wang, Jovana Mitrovic, Frederic Besse, Ioannis Antonoglou, and Lars Buesing. Causally correct partial models for reinforcement learning. *arXiv preprint arXiv:2002.02836*, 2020.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592, 2020.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10740–10749, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Archana Warrier, Thanh Dat Nguyen, Michelangelo Naim, Moksh Jain, Yichao Liang, Karen Schroeder, Cambridge Yang, Joshua B. Tenenbaum, Sebastian Josef Vollmer, Kevin Ellis, and Zenna Tavares. Benchmarking world-model learning. *arXiv preprint arXiv:2510.19788*, 2025.

Appendix

A Additional result tables

Table A.1: Conceptual distinction from reward-free derived-test benchmarks such as WorldTest/AutumnBench. The comparison is about what claims the protocol can identify, not which simulator is more realistic.

Axis	Derived-test protocol	MapShift CPE protocol
Changed factor	Scores a downstream test after reward-free interaction; the changed factor need not be declared as an estimand.	Declares metric, topology, dynamics, or semantic intervention family and severity.
Matched pair	May compare performance across derived tasks or generated worlds.	Holds motif, seed, start state, and non-target factors fixed across e and e'.
Failure attribution	A low score can indicate weak memory, weak update, weak planning, or task mismatch.	Family-wise estimands localize failures to declared environment factors.
Protocol ablations	Not necessarily tied to a same generated intervention grid.	Same-environment, no-exploration, and horizon controls are run on the same grid.
Validity checks	Depends on benchmark-specific documentation.	Emits intervention validators, leakage checks, rejection counts, solvability, and saturation gates.

Table A.2: Benchmark health summary for the expanded release preflight. Health checks are generated before model-result interpretation and included in the release artifact.

Check	Value
Environments and splits	24 motifs; 10 train, 6 validation, 8 test
Task coverage	3456 tasks; 0 task rejections
Reference solvability	1.000
Weak-baseline health estimate	0.415 success proxy; not primary CPE score
Fatal leakage / benign warnings	0 / 0
Intervention validator failures	0
Severity-magnitude failures	0

Table A.3: MapShift release values. These values define the main study and artifact contract.

Component	Value	Component	Value
Map size	96×96	Exploration budget	$T_{\text{exp}} = 800$
Motifs	24	Splits	10 train, 6 val, 8 test
Families	metric/topology/dynamics/semantic	Severities	0, 1, 2, 3
Task classes	planning/inference/adaptation	Task samples	3 per class/cell
Trace-trained seeds	5 per learned baseline	Bootstrap	1000 grouped resamples

Table A.4: Protocol sensitivity for mixed family-primary scores. Ranking comparisons are computed from the same generated 24-motif study grid while changing one evaluation assumption.

Comparison	Kendall τ	Rank reversals	Family changes
No exploration vs. reward-free exploration	1.000	0	1
Same-environment vs. CPE	1.000	0	2

This table uses mixed family-primary scores aggregated over planning, inference, and adaptation; task-conditioned planning slices are analyzed separately in Section 8 and Table 4.

Table A.5: Held-out motif consistency for the mechanism diagnostic. Deltas are classical belief update minus stale-map family scores, averaged over the 8 held-out test motifs. Reversal/reduction counts how often CPE increases the belief-update advantage relative to same-environment evaluation.

Family	Motifs	Mean CPE gap	Mean same-env. gap	Mean protocol delta	BU wins / reduction
Topology	8	0.336	0.102	0.234	8/8; 7/8
Semantic	8	0.724	0.102	0.622	8/8; 8/8

Table A.6: Severity-response summary from the full run. Entries are family primary scores for two calibration baselines across severities 0–3. The severity health gate validates monotone intervention magnitudes, not monotone empirical scores; full method-by-family severity data and rendered curves are included in the artifact.

Baseline	Family	S0	S1	S2	S3
Privileged planning ref.	Metric	0.521	0.486	0.471	0.443
Privileged planning ref.	Topology	0.521	0.552	0.750	0.740
Privileged planning ref.	Dynamics	0.521	0.760	0.792	0.781
Privileged planning ref.	Semantic	0.521	0.802	0.833	0.823
Weak heuristic	Metric	0.512	0.440	0.409	0.407
Weak heuristic	Topology	0.486	0.566	0.444	0.516
Weak heuristic	Dynamics	0.512	0.677	0.697	0.693
Weak heuristic	Semantic	0.486	0.122	0.122	0.141

Table A.7: Family-wise CPE calibration results on non-identity severities 1–3. Entries are mixed primary family scores with equal severity and task-class weights, not planning-reference-normalized upper-bound scores. Deterministic/reference rows use the expanded 24-motif mechanism diagnostic. Learned rows are capacity-calibration rows and are not used for the held-out stale-map versus belief-update claim. The weak heuristic is a hand-coded floor with local spatial priors; full 95% grouped-bootstrap confidence intervals are included in the artifact JSON and rendered Markdown tables.

Method	Metric	Topology	Dynamics	Semantic
Privileged planning ref.	0.439	0.698	0.656	0.761
Weak heuristic	0.375	0.489	0.488	0.167
Classical belief update	0.485	0.689	0.648	0.810
Pretrained graph world model	0.295	0.327	0.400	0.401
Persistent memory	0.313	0.332	0.323	0.354
Relational graph	0.252	0.261	0.252	0.204
Structured dynamics	0.259	0.242	0.280	0.218

Table A.8: Family-wise mechanism diagnostic scores under CPE on non-identity severities 1–3. Scores are mixed primary family scores with equal severity and task-class weights. Planning-only reference path length and reference-gap metrics are tracked separately in the artifact. The diagnostic separates stale pre-intervention reuse, local heuristic priors, and explicit belief update.

Method	Metric	Topology	Dynamics	Semantic
Privileged planning ref.	0.439	0.698	0.656	0.761
Stale-map planner	0.424	0.395	0.656	0.178
Weak heuristic	0.375	0.489	0.488	0.167
Classical belief update	0.485	0.689	0.648	0.810

B Baseline metadata

Table A.9: Deterministic reference baselines. These methods have no trainable parameters and are included to calibrate solvability, protocol sensitivity, and weak-reference behavior.

Baseline	Access and stored state	Evaluation behavior	Training
Privileged post-intervention planning reference	Privileged access to the evaluation environment for solvability and planning-reference checks.	Deterministic shortest-path planning, exact inference answers, and reference adaptation curve when solvable.	0 params; none.
Same-environment reference condition	Protocol-control wrapper used only when execution is performed in the explored environment rather than the intervened environment.	Shares deterministic planning mechanics with the corresponding evaluation condition; interpreted as a protocol comparison condition.	0 params; none.
Weak heuristic	Stores visited cells, visited node ids, visible goal-token bindings, and limited local traversable structure. No global map revision.	Orthogonal visible-path planner over remembered structure; always predicts no topology change; answers semantic reachability with stale remembered token bindings; predicts masked family by task-type rule; capped heuristic adaptation curve.	0 params; none.
Stale-map planner	Stores the serialized pre-intervention environment from reward-free exploration and deliberately does not update it after intervention.	Plans shortest paths and answers semantic queries using the pre-intervention map, then is scored in the evaluation environment. Used as a diagnostic sensitivity floor for stale exploration-derived state.	0 params; none.
Classical belief update	Stores an occupancy map, edge structure, dynamics signature, and semantic-token bindings from exploration.	Uses local post-shift mismatch observations and the task/adaptation budget to update edges, costs, dynamics, and token bindings, then replans. It does not receive the intervention family label or global post-intervention map.	0 params; none.

Training source. Persistent-memory, relational-graph, and structured-dynamics learned baselines are trained separately for each base environment and model seed from the current exploration instance. The pretrained graph world model is a larger capacity-control run trained on generated graph instances before CPE evaluation. In all cases, evaluation-time tensors are built from the exploration-derived graph: node features contain normalized position, local free/blocked/semantic observation statistics, role indicators, and global dynamics features; pair labels supervise edge existence, normalized geometric path cost, and normalized traversal cost; node labels supervise semantic-token location.

Optimization and hyperparameters. Relational and structured-dynamics learned baselines use Adam with learning rate 0.01, validation fraction 0.2, and early-stopping patience 3. The large persistent-memory run uses 2048 memory slots, readout width 672, Adam learning rate 0.0005, 120 epochs, and patience 20. The pretrained graph world model uses Adam with learning rate 0.0005, 120 epochs, and patience 20. Hyperparameters are specified in configuration files before final evaluation; validation motifs are the only intended place for pre-final configuration choices. Test motifs are not used for hyperparameter selection. Checkpoints are keyed by baseline name, environment id, model seed, and hyperparameter hash, then reused across protocol variants.

Compute reporting. The smoke path is designed to run on CPU and completes in approximately 2–5 minutes for the smoke build itself; the one-command artifact audit takes longer because it also runs the test suite. The expanded 24-motif full study is intended for a single CUDA GPU; on an NVIDIA L4 GPU with 23GB memory, reviewers should budget roughly 30–36 wall-clock hours, with faster completion expected on L40S/H100-class GPUs. Preliminary/debug runs are excluded

from the main reported compute. The learned training jobs are small; wall time is dominated by task generation, planning/evaluation loops, bootstrap aggregation, and artifact rendering.

Table A.10: Learned baseline metadata. Parameter counts are representative realized trainable parameters for the release configuration; exact per-run counts, device metadata, checkpoints, and training summaries are written to run manifests.

Baseline	Stored representation and architecture	Supervision and optimization	Size
Pretrained graph world model	Larger graph message-passing model with hidden size 256, 6 message-passing steps, pair width 256, and dynamics width 128.	Pretrained on 4000 generated training graphs with 800 validation graphs, then evaluated under the same CPE task grid; 120 epochs; Adam lr 0.0005; patience 20; 2592 non-identity evaluation episodes.	$\sim$1.14M params.
Persistent-memory world model	Builds the same graph data, but reads each node through learned memory slots. Slot attention stores persistent exploration-derived state.	Large run with 2048 memory slots and readout width 672; same self-supervised heads; 120 epochs; Adam lr 0.0005; validation fraction 0.2; patience 20; 2592 non-identity evaluation episodes.	$\sim$1.4M params.
Relational graph world model	Uses node features and the explored adjacency matrix; applies a node encoder plus 2 message-passing steps over normalized adjacency.	Same self-supervised targets; 10 epochs; Adam lr 0.01; validation fraction 0.2; patience 3; 5 seeds.	396 params.
Structured-dynamics world model	Factorized geometry and dynamics encoders; edge and geometry heads use geometry factors, while traversal-cost prediction additionally receives global dynamics factors.	Same self-supervised targets; 10 epochs; Adam lr 0.01; validation fraction 0.2; patience 3; 5 seeds. Structured by restricting which latent factors feed each prediction head.	504 params.

C MiniGrid adapter contract

This appendix makes the substrate-agnostic claim operational. The artifact includes a MiniGrid-compatible adapter that keeps the CPE interface fixed while replacing the current map generator with a MiniGrid-style environment wrapper. The adapter implements deterministic base sampling, declared intervention application, pair validation, optional instantiation of an actual `minigrid` environment, and a smoke command. The output schema follows MapShift: split id, base environment id, intervention family, severity, rejection reason if any, reference-solvability status, and validator records. The point is not to make MiniGrid look like MapShift, but to make a richer substrate satisfy the same measurement contract.

Table A.11: MiniGrid adapter contract. The included adapter implements the symbolic-grid column; visual or language-conditioned wrappers would add the redesign checks in the right column.

Component	MiniGrid instantiation	Validators that transfer	Validators needing redesign
Base generator	Seeded rooms, doors, keys, walls, lava, and goal objects; motif id replaces map motif.	Split leakage, motif counts, reachable-state coverage, reference solvability.	Mission-template leakage if natural-language missions are used.
Metric shift	Change action cost, step budget, turn/forward cost, or grid-edge weights without changing walls or object identities.	Topology preservation, object identity preservation, path-cost monotonicity.	Cost semantics must be documented because MiniGrid has discrete actions rather than metric distance.
Topology shift	Add/remove doors, walls, blockers, or one-way passages while preserving object identities and start/goal distribution.	Connectivity delta, non-target semantic invariance, reference solvability, task rejection counts.	Door/key dependencies require checking both geometric and inventory-conditioned reachability.
Dynamics shift	Introduce slip, stochastic action failure, locked-door transition changes, or lava penalty changes.	Geometry and semantic preservation, severity-response checks, weak-baseline non-saturation.	Stochastic transitions require repeated rollouts or analytic transition matrices for reference scoring.
Semantic shift	Remap goal color/object type, key-door binding, or mission target while preserving geometry.	Geometry/topology preservation, target-binding change, non-vacuous task checks.	Language templates must be checked so the answer is not leaked by wording.
Tasks	Post-shift navigation, changed-door/key inference, masked-cell prediction, and limited interaction followed by replanning.	Family/task coverage, horizon distributions, final-release rejection accounting.	Observation wrappers must specify whether agents receive symbolic grids, pixels, missions, or both.

Table A.12: MiniGrid-compatible sanity-study result. This is a symbolic MiniGrid-style grid wrapper with optional `minigrid` instantiation, not a pixel/VLA evaluation. The same deterministic mechanism diagnostic is run as in Section 8: privileged planning reference, stale-map planner, weak local planner, and classical belief update under CPE, same-environment, and no-exploration controls.

Quantity	Result
Grid and splits	6 MiniGrid-style motifs, 24 environments; train/val/test = 12/4/8.
Matched intervention pairs	384 pairs across metric, topology, dynamics, and semantic families.
Health checks	0 task rejections, 0 validation failures, reference solvability 1.000.
Weak baseline	Weak local planner score 0.496; non-saturating.
Protocol-sensitive gaps	Belief-update minus stale-map gap under CPE is 0.313 for topology and 0.750 for semantic shifts; same-environment and no-exploration gaps are 0.000 in those controls.
Ranking effect	Semantic same-environment vs. CPE has Kendall $\tau = 0.667$ with one rank reversal.

The same acceptance gates would apply before model results are interpreted: fatal leakage must be zero; final-release task rejection must be reported by cell; the privileged planning reference must solve the released tasks; the weak heuristic must not saturate; severity curves should be interpretable within family; and intervention validators must show that non-target factors are preserved. A ProcTHOR or Habitat port would use the same adapter contract, replacing MiniGrid's grid graph with a navigation graph or navmesh and replacing symbolic-object validators with object-instance, scene-layout, and rendering/visibility validators. This is why the current implementation is useful as a reference artifact: it provides executable tests for the wrapper, not merely a set of benchmark scores.

D Artifact contents

The release artifact is an executable benchmark and generator rather than a static dataset. It contains the code, versioned configs, schemas, and reproduction commands needed to regenerate split manifests, intervention and task recipes, benchmark health reports, per-episode evaluation records, family-wise tables, severity-response data, protocol-comparison outputs, rendered paper tables/figures, and provenance manifests. If generated outputs are additionally submitted or hosted as dataset-like assets, the release package should include Croissant core and Responsible-AI metadata for those assets, while the executable benchmark remains the object needed to inspect the scientific claims.

E Broader impact and ethics

MapShift is an evaluation artifact for embodied-planning research. It does not contain human-subject data, scraped web data, personally identifiable information, or deployed decision systems. The main risk is misuse of benchmark scores as evidence for real-world autonomy or robotics safety beyond the benchmark assumptions. The paper mitigates this by stating supported and unsupported claims, reporting benchmark health separately from model results, and emphasizing family-wise diagnostics over a single leaderboard score.

F NeurIPS Paper Checklist

1. **Claims.** Yes. The abstract and introduction state that MapShift is an executable post-intervention evaluation protocol and benchmark, with claims limited to controlled post-intervention embodied world-model evaluation.
2. **Limitations.** Yes. Section 9 states the supported scope and limitations for interpreting benchmark results.
3. **Theory, assumptions, and proofs.** N/A. The paper formalizes an evaluation protocol and estimands but does not present theoretical theorems requiring proof.
4. **Experimental result reproducibility.** Yes. Methodology, artifact, and appendix sections describe versioned configs, task generation, baseline training, grouped bootstrap estimation, and release artifacts.
5. **Open access to data and code.** Yes. The submission includes an executable artifact containing the benchmark code, versioned configs, schemas, and reproduction commands needed to regenerate the main study results. Section 7 summarizes the smoke-test, audit, main-result reproduction, and MiniGrid-adapter paths.
6. **Experimental setting/details.** Yes. The paper specifies splits, exploration budget, intervention families and severities, task classes, baseline mechanisms, hyperparameters, model seeds, and aggregation.
7. **Experiment statistical significance.** Yes. The paper specifies 95% bootstrap confidence intervals with 1000 resamples grouped by environment/model-seed units.
8. **Experiments compute resources.** Yes. Appendix B reports the expanded full-study setting: a single CUDA GPU, with roughly 30–36 wall-clock hours budgeted on an NVIDIA L4 GPU. It also reports the CPU smoke-test setting and clarifies that preliminary/debug runs are not included in the main reported compute.
9. **Code of ethics.** Yes. The work is designed as a controlled evaluation artifact and does not involve human-subject data, private data, or high-risk deployed systems.
10. **Broader impacts.** Yes. The appendix discusses the main risk: over-interpreting benchmark performance as real-world autonomy readiness.
11. **Safeguards.** N/A. The paper does not release a high-risk pretrained model or dual-use generative model.
12. **Licenses.** Yes. The released code is licensed under the MIT License; generated benchmark outputs and rendered result artifacts are licensed under CC BY 4.0; third-party dependencies and referenced tooling are listed in `THIRD_PARTY_LICENSES.md`.
13. **Assets.** Yes. The paper documents the released executable benchmark artifact, its output schema, limitations, and intended use. No static dataset is introduced; if generated outputs are hosted as dataset-like assets, they should be accompanied by Croissant/RAI metadata.
14. **Crowdsourcing and human subjects.** N/A. The work uses no crowdsourcing and no human subjects.
15. **IRB approvals.** N/A. No human-subject research is conducted.
16. **Declaration of LLM usage.** N/A. LLMs are not a component of the benchmark methodology, baselines, or experimental results.