World Model Triathlon: Executed-Outcome Evaluation of Latent World Models, and a Preregistered Negative Result for LeWM on Push-T

Saloni Garg

Working paper · Draft v0.1 · November 2026

Abstract

World models are usually compared on prediction quality: how well a rollout matches a recorded future, in pixels or in a latent space. That is not the quantity a planner depends on. A planner needs the actions the model ranks first to work when they are executed. This paper describes the World Model Triathlon, a small preregistered protocol that scores world models on executed outcomes only. Models face the same initial states, the same goals and the same planner budget, and the score is what the simulator records after the chosen actions are executed, never the model's own predicted outcome. The protocol was frozen before any run, and every later change is recorded as a dated amendment. We report the first completed runs, all for one latent model, LeWM, on one task, Push-T. On Leg A, an open-loop ranking test over 100 shared states, LeWM's top-ranked candidate ended with a mean executed state error of 307.33, worse than the mean initial distance to the goal of 272.3, with executed success on 1 of 100 states. On Leg B, a closed-loop planning test over 50 goal episodes, LeWM solved 0 of 50 episodes at a short horizon, and 0 of 50 again when the horizon was lengthened by a factor of 6 with real replanning under a pre-committed amendment. The longer run ended farther from the goal on average (mean error 778.25) than the short one (363.56). We report this as what it is: a negative result for one model on one task, under a stated protocol, at a total spend of $0. The scope cuts that shaped the result, including the removal of the generative and RL baselines for budget reasons, are printed in the paper rather than left to a footnote.

1. Introduction

A world model has two different jobs that are easy to confuse. The first is prediction: given a state and an action sequence, produce something close to what happens next. The second is control: rank candidate action sequences so that executing the best-ranked one moves the world toward a goal. Benchmarks mostly measure the first job. A model can score well at prediction and still rank candidates badly, because ranking only needs the relative order of candidates to be right near the top, and because small prediction errors compound once actions are executed and the state moves off the distribution the model was scored on.

The Triathlon protocol measures the second job directly. Its rules are simple:

- Every model gets the same initial states and the same goals, generated once and shared byte-for-byte.
- Every model that plans with the cross-entropy method (CEM) gets the same frozen planner budget. No per-model tuning is allowed.
- The score is computed from the state the simulator reaches after execution. Model-predicted outcomes are recorded separately where a model exposes them, and are never mixed with executed outcomes.
- Compute is accounted per row: wall-clock time and device time are part of the result, not an afterthought.

- Exclusions are defined in advance and counted. No episode is excluded because a model did badly on it, and no seed is re-drawn after results are seen.

The protocol was preregistered and frozen on 2026-11-22, before any Triathlon run. The original design raced three families (latent JEPA models, a generative video model and an RL world model) across three legs: Leg A (ranking of shared candidates), Leg B (closed-loop planning) and Leg C (counterfactual pairs). Section 3 records, in the open, how a budget decision narrowed that design before most of it ran. What survives is narrower than the original claim, and this paper claims only what was run.

The contributions are:

1. A frozen, citable executed-outcome protocol with a results schema, a validator and a generated leaderboard (Section 2).
2. A complete first set of runs for LeWM on Push-T, including adapter validity controls that separate model behaviour from wiring errors (Section 4).
3. A preregistered negative result, plus a pre-committed follow-up test that removed the most obvious excuse for it: the horizon was lengthened by a factor of 6, with replanning, and the result did not change (Sections 4 and 5).

2. Protocol

2.1 Tasks and legs

The frozen task set v1 has two tasks: Push-T (stable-worldmodel environment `swm/PushT-v1`) and OGBench-Cube (`swm/OGBCube-v0`). Push-T is the task used here. It was chosen because every latent baseline in the original design publishes a Push-T checkpoint, so a failure on Push-T is informative rather than a setup artefact. LeWM's own published Push-T result under its own evaluation protocol is about 94% success in-distribution; the gap between that number and the numbers in Section 4 is part of what this protocol exists to expose, since the two numbers are produced by different scoring rules on different state sets.

- **Leg A (ranking).** 100 initial states, seeds 0-99. The model ranks a shared candidate list of action sequences; the runner executes its top pick. Score: mean Euclidean error between the executed final state and the goal state, in the simulator's native state units, plus the executed success count from the environment's own flag. For LeWM, whose step unit is a 5-environment-step action block baked into the checkpoint, the frozen horizon of 10 was executed as 10 model steps, that is 50 executed environment steps. This reading matches the step mapping used by stable-worldmodel's own LeWM evaluation and is applied uniformly to any future Leg A baseline.
- **Leg B (planning).** 50 goal episodes, seeds 1000-1049. The model plans with CEM toward the episode goal and executes under a receding horizon. Score: executed success rate from the environment's own flag, mean executed state error, mean wall-clock per episode, and cost per solved episode (total cost divided by successes; reported as n/a when there are no successes, never as zero).
- **Leg C (counterfactual pairs).** Defined in the preregistration. Not run. No Leg C numbers appear in this paper.

2.2 Planner budget

The frozen CEM budget, identical for every model that uses CEM, is: 300 samples, plan horizon 10 model steps, receding horizon 5 model steps, with iteration counts as in the stable-worldmodel WorldModelPolicy default. The LeWM checkpoint used throughout is the public Push-T checkpoint `quentinll/lewm-pusht`, loaded through stable-worldmodel 0.1.1. Checkpoint selection followed the preregistration: the named public checkpoint, with no search across a checkpoint suite.

2.3 Scoring, exclusions and citation

Executed state error is the environment's own `eval_state` distance between the executed final state and the goal state. Success is the environment's own flag at the end of the executed horizon. An episode is excluded only if the simulator errors or returns a non-finite state, the initial state file is corrupt, or the model process crashes before producing any action. A crash after at least one executed action counts as a failure, not an exclusion. All three runs reported here had 0 exclusions.

Every score in this paper belongs to the specification citation:

`WorldModelTriathlon-0.1 / protocol=1.0 / taskset=pusht-v1+ogbcube-v1`

A score reported without its run id is not a Triathlon score. The three run ids reported here are `gate1-lewm-pusht-leg-a`, `amend1-lewm-pusht-leg-b` and `amend2-lewm-pusht-leg-b`. Amendment runs never overwrite or merge with the rows they amend; they are reported as separate runs and compared explicitly. Results are stored as one JSON row per episode, checked against a published schema with a validator, and the leaderboard is generated from the rows, never hand-edited.

2.4 Preregistered claims, and their status

The preregistration stated a primary claim (P1: the executed-outcome family ranking reverses the video-quality ranking, with latent JEPA planners beating the generative baseline on executed success per GPU-second) and secondary claims on the Leg A versus Leg B dissociation (S1), counterfactual pairs (S2) and the cost gap between families (S3). After Amendment 1 (Section 3), P1 as written cannot be tested, because the generative baseline it names was cut. This paper therefore does not test P1, S2 or S3. It reports the within-model Leg A versus Leg B comparison relevant to S1 for a single model, which cannot by itself confirm or refute a cross-model correlation claim. We state this plainly because a preregistration that is quietly re-read after the fact is worth nothing.

3. Scope decisions and amendments

Two amendments were approved after the freeze and before the affected runs. Both are part of the record.

Amendment 1 (2026-11-23): a $100 total budget cap. The project has no larger budget; the total cap is $100 and the total spend to date is $0, with every run executed on local CPU. Consequences, printed here as decided:

- Cosmos Predict, the generative baseline, was cut. Even its capped Leg A could not be run honestly within the cap.
- The Meta-World subset was cut. It was already conditional in the preregistration and needs a separate software stack.
- TD-MPC2 and V-JEPA 2-AC were deferred, not cut outright: each may be added only after a separate cost estimate is shown and approved. Neither has been run, and no numbers for either appear in this paper.

- The headline claim narrows from a cross-family race to the latent-family executed race plus the Leg A versus Leg B comparison. DINO-WM and PLDM remain in scope at $0 local cost if their checkpoints can be obtained at no cost; they have not been run yet.

Amendment 2 (2026-11-23): Leg B horizon lengthened from 25 to 150 executed steps. The reason is in Section 4.2: at 25 executed steps each episode contained exactly one CEM solve, so the first Leg B run tested single-plan execution, not replanning, and 25 steps is far below the Push-T task's normal episode length. The rerun kept the same 50 episodes, the same seeds and goals, the same CEM budget and no tuning of any kind. Before the rerun started, the interpretation was committed in writing: if the rerun also gives near-zero success while ending farther than the start, the short-horizon explanation for the first run's zero is largely removed; if success jumps, the first zero was a horizon artefact and must be reported as one. Section 5 follows that commitment.

One further clarification from the first run is recorded in Amendment 1: the Leg A horizon mapping for LeWM (10 model steps = 50 executed environment steps), described in Section 2.1.

4. Results

All runs: LeWM, Push-T, local CPU only, $0 spent, 0 exclusions. State error is in the simulator's native units; lower is better.

4.1 Leg A: ranking (run `gate1-lewm-pusht-leg-a`)

Over the 100 shared states (seeds 0-99), each with a shared list of 32 candidate action sequences generated once and saved with hashes:

Quantity	Value
LeWM mean executed state error	307.33
Median (min, max)	328.84 (21.39, 659.96)
Executed success	1 / 100
Mean initial distance to goal (no action)	272.3
Simulator-oracle candidate, mean executed error	145.6 (success 2 / 100)
One fixed random candidate, mean executed error	458.7
Total wall time, 100 states	210.8 s

LeWM's top-ranked candidate ended farther from the goal on average (307.33) than the initial state itself (272.3). In other words, at this horizon and under this scoring, executing LeWM's first choice was worse on average than doing nothing, and far behind the oracle candidate sitting in the same shared list (145.6). LeWM's pick came from the oracle block of the candidate list in 41 of 100 states and from the random block in 59 of 100.

Before the full run, adapter validity controls were run on one state to check that the harness reports the model's own costs faithfully. With the goal set to the initial frame, a zero-action candidate scored a cost of 15.1 against 186.9 for a random candidate. With the goal set to the actual final frame of the oracle trajectory, the oracle candidate scored 25.9 against 231.4 for random and 116.7 for zero action. At a native 5-step horizon toward a 25-step goal frame, the oracle scored 9.8 against 35.1 for random. The adapter orders these controls correctly, so the weak ranking above is the model's behaviour under the protocol, not a wiring artefact. As a further check, seeds 0,

47 and 99 were re-executed independently after the run; success and state error matched the recorded rows exactly, 3 of 3.

This is a statement about open-loop ranking at a fixed horizon. It is not, by itself, a statement about LeWM's closed-loop planning, which replans as it executes. Leg B tests that.

4.2 Leg B, version 1: closed loop at 25 executed steps (run `amend1-lewm-pusht-leg-b`)

Over the 50 goal episodes (seeds 1000-1049), executed horizon 25 environment steps:

Quantity	Value
Executed success	0 / 50
Mean executed state error	363.56
Median (min, max)	310.72 (124.40, 1162.33)
Mean initial distance to goal	248.67
Episodes ending closer / farther than they started	15 / 35
Mean wall time per episode	214.9 s
Cost per solved episode	n/a (0 successes)

Closed-loop control did not beat open-loop ranking here: 0 of 50 against Leg A's 1 of 100, with executed error in the same range or worse, and episodes ending farther from the goal than they started on average (mean change +114.88). Two caveats belong with this comparison, and both were recorded at the time. First, the two legs use different state sets and different executed horizons (25 against 50 environment steps), so this is not a paired comparison. Second, and more important: at the frozen receding horizon, one episode of 25 executed steps contains exactly one CEM solve. Version 1 therefore tested plan execution, not replanning. A zero from version 1 alone would not separate model failure from horizon insufficiency, since the simulator-oracle candidate in Leg A also reached only 2 of 100 at a short horizon. That is the gap Amendment 2 was written to close.

A spot check re-executed 3 episodes with a fresh solver. Success agreed on all 3 (all failures). The recorded error for seed 1000 reproduced exactly; the errors for seeds 1025 and 1049 differed on re-execution (215.31 recorded against 212.95 re-executed, and 572.35 against 686.87). The run had been restarted twice after the runtime environment killed the process, and resumed segments start a fresh solver, so the CEM sampling stream differs across segments. The success verdicts are stable under this; individual error values for a given seed carry that variance. Section 6 returns to this.

4.3 Leg B, Amendment 2: closed loop at 150 executed steps (run `amend2-lewm-pusht-leg-b`)

The rerun changed one thing: the executed horizon, from 25 to 150 environment steps (30 model steps, 6 CEM solves per episode at the unchanged receding horizon). Same 50 episodes, seeds, goals, CEM budget and scorer; the initial states and goal states are identical to version 1 for all 50 seeds, so the comparison is paired.

Quantity	v1 (25 steps, 1 solve)	v2 (150 steps, 6 solves)
Executed success	0 / 50	0 / 50
Mean executed state error	363.56	778.25

Quantity	v1 (25 steps, 1 solve)	v2 (150 steps, 6 solves)
Median state error	310.72	421.39
Min / max error	124.40 / 1162.33	75.94 / 5929.50
Mean initial distance	248.67	248.67
Mean change (final minus initial)	+114.88	+529.57
Ended closer / farther	15 / 35	14 / 36
Mean wall time per episode	214.9 s	1744.7 s

With 6 times the executed horizon and real replanning, LeWM still solved 0 of 50 episodes, and ended farther from the goal than it started, on average and by median. Paired per seed, the version 2 error was lower than version 1 on 21 seeds and higher on 29; the mean paired difference was +414.69 and the median paired difference was +121.90. The version 2 mean is pulled up by two large outliers (seed 1030 at 5929.50 and seed 1022 at 4763.65; standard deviation 1091.05), but the median also rose from 310.72 to 421.39, so the worsening is not only an outlier effect. The longer horizon did not rescue planning; it gave the planner more steps in which to move the block away from the goal.

Verification for this run was by row audit rather than full re-execution. All 200 project rows (100 Leg A, 50 Leg B v1, 50 Leg B v2) were recomputed from the results file: the Amendment 2 seeds are exactly 1000-1049, each once, with 6 solves and 150 executed steps recorded on all 50 rows, device CPU and GPU-seconds 0.0 throughout. The recorded state error for every Amendment 2 row was reproduced from its recorded final and goal states to a maximum difference of 2.3e-13. A full re-execution spot check was not run for this run: one episode costs roughly 20-40 minutes of CPU, and version 1 showed that re-execution after restarts checks the pipeline rather than the recorded number, because CEM outcomes shift with the solver's random stream position. The recorded numbers are internally exact; their seed-level reproducibility is limited in the way Section 6 describes.

4.4 Summary of the three runs

Run	Leg	N	Success	Mean executed state error
`gate1-lewm-pusht-leg-a`	A (ranking, 50 steps)	100	1 / 100	307.33
`amend1-lewm-pusht-leg-b`	B (25 steps, 1 solve)	50	0 / 50	363.56
`amend2-lewm-pusht-leg-b`	B (150 steps, 6 solves)	50	0 / 50	778.25

5. Interpretation

The pre-committed reading of Amendment 2 was written before the rerun: a second zero, with episodes ending farther than they started, would mean the first zero was not mainly a horizon artefact; a jump in success would mean it was. The data supports the first branch. Version 1's 0 of 50 should no longer be read as mainly a too-short-horizon artefact.

Taken together, the three runs support a narrow statement and no more: under this protocol, on Push-T, LeWM's executed outcomes are poor in both modes tested. Its one-shot ranking picks end farther from the goal than the start state on average, and its closed-loop CEM planning, given a normal-length horizon and repeated replanning,

solves none of 50 episodes and moves the block away from the goal more often than toward it. This sits next to LeWM's published in-distribution Push-T success of about 94% under its own evaluation protocol. The two results are not in arithmetic conflict; they are answers to different questions, asked under different scoring rules, state sets and horizons. The point of an executed-outcome protocol is that the difference between those questions is itself a finding worth printing, because deployment depends on the executed answer.

Nothing here ranks model families. One model was run. The cross-family claims in the preregistration remain untested, and the dissociation claim S1, which is a claim about a correlation across models, cannot be settled by one model's two legs. What the single-model result does is remove one comfortable explanation (too short a horizon) for one model, and leave the protocol, schema and leaderboard ready for the models that are still in scope.

6. Threats to validity

- **One model, one task.** Every number in this paper is LeWM on Push-T. Nothing follows about latent world models in general, about LeWM on other tasks (including OGBench-Cube, which is in the frozen task set but has not been run), or about any other family. The title says so, and the claim should not be quoted more widely than that.
- **Protocol distance from the model's own evaluation.** The Triathlon state sets, candidate lists, horizons and scoring are not LeWM's published evaluation conditions, and the gap to the published 94% figure is partly a difference of conditions by design. The protocol is uniform and stated, which is what makes the comparison fair across future models; it does not make the conditions neutral for this one.
- **CPU only, and slow.** All runs were executed on a local CPU virtual machine with no GPU, at $0. Mean wall time reached 1744.7 s per episode in the Amendment 2 run. CPU execution should not change executed outcomes for a fixed solver stream, but it limited how much could be run and ruled out full re-execution checks for the long run.
- **CEM sampling variance and restarts.** The Leg B runs were interrupted by the runtime environment and resumed; each resume starts a fresh solver, which shifts the CEM sampling stream. In the version 1 spot check, success agreed on 3 of 3 re-executed episodes but two of three error values moved, one of them substantially (572.35 recorded against 686.87 re-executed). Seed-level error values should therefore be read as samples from a solver distribution, not as exact constants. The aggregate picture (0 of 50 in both versions, errors in the hundreds) is not sensitive to this variance; a finer claim would be.
- **Outlier sensitivity of the mean.** The Amendment 2 mean error of 778.25 is pulled up by two episodes (5929.50 and 4763.65) and has a standard deviation of 1091.05. The median (421.39) is the more robust summary, and it also rose relative to version 1 (310.72), which is why we report the worsening as real while flagging the mean.
- **Horizon mapping.** Leg A executed 10 model steps as 50 environment steps because the checkpoint's step is a 5-step action block. This follows the model's own evaluation convention, but a reader who reads "horizon 10" as 10 environment steps will misread Leg A. The mapping is stated wherever the Leg A numbers appear.
- **Scoring depends on the environment's own functions.** Success and state error are the simulator's own flag and `eval_state` distance. That is a deliberate choice (one scorer for every model), but the results inherit whatever those functions reward.

- **No Leg C, no second task, no second model yet.** The dissociation and counterfactual parts of the preregistration are untested. This paper is an interim report on a frozen protocol, not the completed Triathlon.

7. Reproducibility

The preregistration, protocol and specification are versioned documents in the project repository, with amendments dated and signed off before the runs they affect. Results are stored as one JSON row per episode in a single results file (200 rows: 100 Leg A, 50 Leg B v1, 50 Leg B Amendment 2), validated against a published schema. Shared Leg A candidate lists were generated once and saved with SHA-256 hashes. The leaderboard is generated from the rows by a script, and a reproduction script asserts the headline numbers in Section 4 from the raw rows. Environment versions are recorded in every row; two pins mattered in practice and are recorded here as in the run summaries: stable-worldmodel 0.1.1, and transformers 4.57.3, because the checkpoint's state dictionary uses older HuggingFace ViT key names that transformers 5.24 cannot load. Actions were z-scored with the training normalisation statistics of the full action column (n = 2,336,736) of the LeWM Push-T dataset, the same scaling source for all three runs. Total spend: $0 of the $100 cap.

8. Conclusion

We built a small, frozen protocol that scores world models only on what happens when their chosen actions are executed, and ran its first model. LeWM, on Push-T, ranked candidates worse than doing nothing on average and solved no closed-loop episode at either of two horizons, the second six times longer than the first with real replanning, under an interpretation committed before the rerun. That is a negative result about one model under one protocol, and we present it at exactly that strength. The protocol, the schema and the leaderboard are the durable part of this work: the remaining in-scope models can now be added at $0 local cost, and when they are, their scores will be comparable to these because the rules were not changed to fit the first result.

References

- LeWorldModel (LeWM): latent joint-embedding predictive world model. Public Push-T checkpoint `quentinll/lewm-pusht` on Hugging Face, used as distributed.
- stable-worldmodel: the library and environment suite used for all runs (galilai-group/stable-worldmodel), including the `swm/PushT-v1` environment, the WorldModelPolicy evaluation path and the CEM solver defaults.
- Push-T: the planar pushing task introduced with Diffusion Policy (Chi et al., Diffusion Policy: Visuomotor Policy Learning via Action Diffusion).
- Cross-entropy method (CEM): the planning optimiser used under the frozen budget in Section 2.2 (Rubinstein, cross-entropy method for optimisation).

Appendix: run record

Item	Value
Protocol frozen	2026-11-22

Item	Value
Leg A run date	2026-11-22
Leg B v1 run date	2026-11-23
Leg B Amendment 2 completed	2026-11-25
Amendment 1 (budget cap $100)	Approved 2026-11-23
Amendment 2 (horizon 25 to 150 steps)	Approved 2026-11-23
Total spend	$0 of $100 cap
Device	Local CPU only, GPU-seconds 0.0 on all rows
Total result rows	200, exclusions 0