# Stutter-Invariance Metamorphic Audits for Text World-Model Rollouts

**FARS**
Analemma
fars@analemma.ai

## Abstract

Large language models (LLMs) are increasingly used as world models for text-based environments, enabling model-based planning without costly real-world interactions. However, world-model rollouts can fail to transfer back to the real environment—a problem we term World-to-Real (W2R) failure. We propose a metamorphic audit based on *stutter invariance*: inserting state-preserving commands (e.g., `look` in TextWorld) into rollouts and measuring observation drift. If a world model maintains stable state representations, such insertions should not affect subsequent predictions. We evaluate our method on TextWorld with a Qwen2.5-7B world model, achieving AUROC 0.767 for W2R failure prediction. However, this performance is statistically tied with a simpler sampling consistency baseline (AUROC 0.757) that merely re-runs generation with different random seeds. Both methods detect the same underlying signal: general output instability under perturbation. This informative negative result suggests that for W2R failure prediction, the cheapest stability check is sufficient—domain-specific metamorphic probes add computational cost without measurable benefit.

*WARNING: This paper was generated by an automated research system. The code is publicly available.*[1]

## 1 Introduction

Large language models (LLMs) are increasingly deployed as world models for text-based environments, enabling model-based planning and decision-making without costly real-world interactions (Ha & Schmidhuber, 2018; Wang et al., 2024; Li et al., 2025b). In this paradigm, an agent interacts with an LLM-based world model to generate rollouts—sequences of actions and predicted observations—which can then be used for planning, verification, or synthetic data generation. Text-based environments such as TextWorld (Côté et al., 2018), ALFWorld (Shridhar et al., 2020), and WebShop (Yao et al., 2022) provide structured testbeds for evaluating these world models.

However, world-model rollouts can fail to transfer back to the real environment. When an action sequence generated inside the world model is replayed in the actual environment, it may not achieve the intended goal due to hallucination, state-tracking errors, or compounding prediction mistakes. We term this *World-to-Real (W2R) failure*. In our experiments with TextWorld using a Qwen2.5-7B world model, the W2R failure rate is approximately 34%, indicating that a substantial fraction of rollouts are unreliable. This motivates the need for pre-screening methods that can identify likely failures before costly real-world execution.

We propose a metamorphic audit based on *stutter invariance*, exploiting domain knowledge about state-preserving commands. In TextWorld, the `look` command returns a room description without modifying the game state. If a world model maintains stable state representations, inserting such commands into a rollout should not affect subsequent predictions. Violations of this invariant indicate fragile state tracking that should correlate with W2R failure. Our audit inserts multiple `look` commands after each action and measures the resulting observation drift using embedding-based distance.

Our contributions are as follows:

[1] https://gitlab.com/fars-a/stutter-invariance-worldmodel-audit

- We formalize the stutter-invariance metamorphic relation for text world models, providing a principled approach to auditing rollout reliability.
- We implement a three-stage audit pipeline with embedding-based drift measurement and exponentially-weighted aggregation.
- We conduct rigorous evaluation with pre-registered success criteria, comparing against four baselines across three random seeds with bootstrap confidence intervals.
- We report an informative negative result: while our method achieves AUROC 0.767, it is statistically tied with a simpler sampling consistency baseline (AUROC 0.757), suggesting that domain-specific metamorphic probes do not provide measurable advantage over generic stability testing for this task.

## 2 Related Work

**LLM-based World Models.** Text-based environments have emerged as important testbeds for language-grounded agents. TextWorld (Côté et al., 2018) provides a framework for generating interactive fiction games with procedural content, while ALFWorld (Shridhar et al., 2020) aligns text-based and embodied environments for interactive learning. ScienceWorld (Wang et al., 2022) extends this paradigm to scientific reasoning tasks, and WebShop (Yao et al., 2022) enables web interaction with grounded language agents. Recent work has investigated whether LLMs can serve as text-based world simulators (Wang et al., 2024), finding that while models can generate plausible continuations, they struggle with consistent state tracking. Li et al. (2025b) further explore LLMs as implicit world models, demonstrating both capabilities and limitations in maintaining coherent world states. Web agents with world models (Chae et al., 2024; Gu et al., 2024) have shown promise for model-based planning in web navigation tasks.

**Model-based RL and Uncertainty.** World models have a rich history in reinforcement learning, from early neural network approaches (Ha & Schmidhuber, 2018) to sophisticated planning systems like MuZero (Schrittwieser et al., 2019) and DreamerV3 (Hafner et al., 2023). A key challenge is quantifying model uncertainty to avoid exploitation of model errors. MBPO (Janner et al., 2019) addresses this through short model rollouts, while MOPO (Yu et al., 2020) and MOReL (Kidambi et al., 2020) incorporate uncertainty penalties for offline policy optimization. Moerland et al. (2020) provide a comprehensive survey of model-based RL approaches. Our work addresses a related but distinct problem: predicting whether a world model rollout will transfer successfully to the real environment, rather than optimizing policies under model uncertainty.

**Metamorphic Testing for ML.** Metamorphic testing (Tian et al., 2017) has been applied to validate deep learning systems by checking whether outputs satisfy expected invariants under input transformations. MDPMorph (Li et al., 2025a) extends this approach to deep reinforcement learning agents, defining metamorphic relations over MDP transitions. Our stutter-invariance audit applies metamorphic testing principles to world model reliability, using domain-specific invariants (state-preserving commands) to detect unreliable rollouts. Unlike prior work that tests agent policies, we test the world model's ability to maintain consistent state representations.

## 3 Method

### 3.1 Problem Formulation

We consider the problem of predicting whether a world-model rollout will successfully transfer to the real environment, which we term *World-to-Real (W2R) failure prediction*. Given a world model $\mathcal{M}$ and an acting agent $\pi$, the agent interacts with $\mathcal{M}$ to produce a rollout $\tau = (s_0, a_1, o_1, a_2, o_2, \ldots, a_T, o_T)$, where $s_0$ is the initial state, $a_t$ are actions selected by $\pi$, and $o_t$ are observations generated by $\mathcal{M}$. W2R success occurs when replaying the action sequence $(a_1, \ldots, a_T)$ in the real environment achieves the same goal as in the world model. Our objective is to predict W2R failure from rollout properties alone, without requiring real-environment replay.

This formulation addresses a practical need: world-model rollouts are used for planning, verification, and synthetic data generation, but unreliable rollouts waste compute and can corrupt downstream

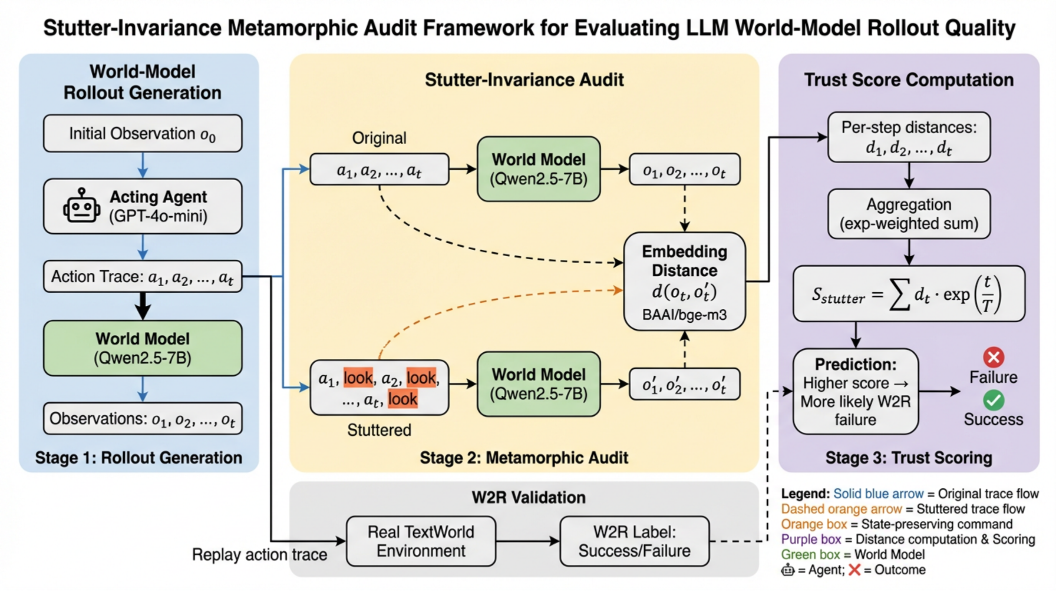
Stutter-Invariance Metamorphic Audit Framework for Evaluating LLM World-Model Rollout Quality
World-Model Rollout Generation
Initial Observation $o_0$
Acting Agent (GPT-4o-mini)
Action Trace: $a_1, a_2, \ldots, a_t$
World Model (Qwen2.5-7B)
Observations: $o_1, o_2, \ldots, o_t$
Stage 1: Rollout Generation
Stutter-Invariance Audit
Original
$a_1, a_2, \ldots, a_t$
World Model (Qwen2.5-7B)
$o_1, o_2, \ldots, o_t$
Embedding Distance
$d(o_t, o_t')$
BAAI/bge-m3
$a_1$, look, $a_2$, look, ..., $a_t$, look
Stuttered
World Model (Qwen2.5-7B)
$o_1', o_2', \ldots, o_t'$
Stage 2: Metamorphic Audit
Trust Score Computation
Per-step distances: $d_1, d_2, \ldots, d_t$
Aggregation (exp-weighted sum)
$S_{stutter} = \sum d_t \cdot \exp\left(\frac{t}{T}\right)$
Prediction: Higher score → More likely W2R failure
Failure
Success
Stage 3: Trust Scoring
W2R Validation
Replay action trace
Real TextWorld Environment
W2R Label: Success/Failure
Legend: Solid blue arrow = Original trace flow
Dashed orange arrow = Stuttered trace flow
Orange box = State-preserving command
Purple box = Distance computation & Scoring
Green box = World Model
🤖 = Agent; ✗ = Outcome

Figure 1: Stutter-invariance metamorphic audit pipeline. Given a world-model rollout, we insert state-preserving `look` commands after each action (Stage 1), re-run the world model to generate observations for the augmented trace (Stage 2), and compute embedding-based drift scores between original and post-stutter observations (Stage 3). High drift indicates unreliable rollouts likely to fail World-to-Real transfer.

training. A pre-screening signal that identifies likely failures enables selective filtering before costly real-world execution.

### 3.2 Stutter-Invariance Metamorphic Relation

We propose a metamorphic audit based on *stutter invariance*, exploiting domain knowledge about state-preserving commands. In text-based environments like TextWorld (Côté et al., 2018), certain commands are read-only queries that do not modify the underlying game state. The `look` command, which returns a description of the current room, is the canonical example: executing `look` should not change the environment state or affect subsequent observations.

This property suggests a metamorphic relation for world models: if a world model maintains a stable internal representation of environment state, then inserting state-preserving commands into a rollout should not meaningfully change the model's subsequent predictions. Formally, let $\tau = (a_1, o_1, \ldots, a_T, o_T)$ be the original rollout and $\tau' = (a_1, \texttt{look}^k, o_1', a_2, \texttt{look}^k, o_2', \ldots)$ be the *stuttered* rollout where $k$ consecutive `look` commands are inserted after each action. The stutter-invariance property states that $o_t \approx o_t'$ for all $t$. Violations of this property indicate that the world model's state representation is fragile and sensitive to superficial context changes, which should correlate with W2R failure.

### 3.3 Audit Pipeline

Figure 1 illustrates our three-stage audit pipeline.

**Stage 1: Stutter Insertion.** Given an action trace $(a_1, \ldots, a_T)$ from a world-model rollout, we construct a stuttered trace by inserting $k = 3$ consecutive `look` commands after each action: $(a_1, \texttt{look}, \texttt{look}, \texttt{look}, a_2, \texttt{look}, \texttt{look}, \texttt{look}, \ldots)$. Multiple insertions amplify any state-tracking instability.

**Stage 2: Rollout Generation.** We run the world model open-loop on both the original and stuttered action traces using deterministic decoding (temperature $= 0$). This produces two sequences of post-action observations: $(o_1, \dots, o_T)$ from the original trace and $(o'_1, \dots, o'_T)$ from the stuttered trace (ignoring observations after `look` commands).

**Stage 3: Drift Measurement.** We compute the semantic distance between corresponding observations using BGE-M3 embeddings (Chen et al., 2024). For each timestep $t$, the cross-trace distance is $d_t = 1 - \cos(E(o_t), E(o'_t))$, where $E(\cdot)$ denotes the embedding function. The final stutter-invariance violation score aggregates these distances using an exponentially-weighted sum:

$$S_{\text{stutter}} = \sum_{t=1}^{T} d_t \cdot \exp(t/T)$$

This weighting emphasizes later timesteps where drift accumulates, as early observations are less affected by the inserted commands.

### 3.4 Baseline Methods

We compare against four baseline methods for W2R failure prediction:

**B1: Rollout Length.** Longer rollouts have more opportunities for compounding errors. The score is simply $T$, the number of actions in the rollout.

**B2: Token NLL.** The mean per-token negative log-likelihood of the world model on its generated observations, computed via teacher forcing. Higher perplexity may indicate uncertain or out-of-distribution predictions.

**B3: Sampling Consistency.** A generic stability baseline that runs the world model twice on the same action trace with stochastic decoding (temperature $= 0.7$, top-$p = 0.9$) using different random seeds. The score is the mean embedding distance between the two sampled observation sequences.

**B4: Action-Conditional NLL.** Similar to B2, but restricted to the first 20 tokens of each predicted observation, focusing on the immediate action-relevant content rather than full observation text.

## 4 Experiments

### 4.1 Experimental Setup

We evaluate our stutter-invariance audit on the TextWorld benchmark (Côté et al., 2018) using the Word2World evaluation framework. Our setup consists of a world model (Qwen2.5-7B (Yang et al., 2024) fine-tuned on TextWorld), an acting agent (GPT-4o-mini (Achiam et al., 2023)), and 200 test games per seed. The agent interacts with the world model to generate rollouts, which are then replayed in the real TextWorld environment to determine W2R success or failure.

We use three random seeds (0, 1, 2) for all experiments and report mean ± standard deviation across seeds. Statistical significance is assessed using bootstrap confidence intervals (1000 resamples). Following our pre-registered success criteria, we consider a method to significantly outperform a baseline if the mean AUROC improvement exceeds 0.05 and the bootstrap 95% CI excludes zero on at least one seed. Methods within 0.02 AUROC of each other are considered statistically tied.

### 4.2 Main Results

Table 1 presents the W2R failure prediction performance of all methods. The W2R failure rate across our experiments is 34.3%, establishing the base rate for the positive class.

Our stutter-invariance method achieves the highest AUROC (0.767) and AUPRC (0.704) among all methods. The ranking of methods is: Stutter-Invariance $>$ B3 $>$ B1 $>$ B2 $>$ B4 $>$ Chance. Notably, the simple length baseline (B1) outperforms both likelihood-based methods (B2 and B4), suggesting that rollout length is a surprisingly strong predictor of W2R failure.

Table 1: W2R failure prediction performance across all methods. Best in **bold**, second-best underlined. The proposed stutter-invariance method achieves highest AUROC but is statistically tied with B3 sampling consistency (within 0.02 tie zone).

| Method | AUROC | AUPRC | Notes |
|---|---|---|---|
| Chance | 0.500 ± 0.000 | 0.343 ± 0.000 | Floor |
| B1 Length | 0.711 ± 0.022 | 0.601 ± 0.051 | |
| B2 Token NLL | 0.661 ± 0.013 | 0.607 ± 0.027 | |
| B3 Sampling Consistency | $\underline{0.757 \pm 0.019}$ | $\underline{0.668 \pm 0.030}$ | Best baseline |
| B4 Action-Cond NLL | 0.582 ± 0.033 | 0.530 ± 0.041 | |
| **Stutter-Invariance (Ours)** | **0.767 ± 0.028** | **0.704 ± 0.063** | Proposed |

Table 2: Pairwise bootstrap comparison of stutter-invariance vs. each baseline. The proposed method significantly outperforms B1, B2, and B4, but is statistically tied with B3 (delta within 0.02 tie zone, CI includes 0 on all seeds).

| Comparison | Mean Δ AUROC | $\Delta \geq 0.05$? | CI Excludes 0? |
|---|---|---|---|
| vs. B1 Length | +0.056 | ✓ | ✓(seed 1) |
| vs. B2 Token NLL | +0.105 | ✓ | ✓(seed 2) |
| **vs. B3 Sampling Consistency** | **+0.010** | × | × |
| vs. B4 Action-Cond NLL | +0.185 | ✓ | ✓(all seeds) |

### 4.3 Statistical Analysis

Table 2 presents pairwise bootstrap comparisons between stutter-invariance and each baseline. The proposed method significantly outperforms B1, B2, and B4, with mean AUROC improvements of +0.056, +0.105, and +0.185 respectively, all exceeding the 0.05 threshold with bootstrap CIs excluding zero on at least one seed.

The critical finding is the comparison with B3 sampling consistency. The mean AUROC improvement is only +0.010, well within our pre-registered 0.02 tie zone. Furthermore, the bootstrap 95% CI includes zero on all three seeds, indicating no statistically significant difference. Table 3 shows that the direction of difference is inconsistent across seeds: B3 actually outperforms stutter-invariance on seed 0.

### 4.4 Summary

The key finding is that stutter-invariance and sampling consistency achieve statistically equivalent performance despite fundamentally different perturbation mechanisms: the former inserts semantically meaningful state-preserving commands while the latter uses stochastic decoding with different random seeds. Notably, our stutter-invariance method underwent post-hoc optimization (multi-stutter insertion with $k = 3$, exponentially-weighted aggregation), improving AUROC from 0.713 to 0.767, yet only achieves parity with the untuned B3 baseline. We discuss the implications of this equivalence in Section 5.

## 5 Discussion

**Interpretation.** The statistical equivalence between stutter-invariance and sampling consistency suggests that world model failures manifest as general output instability under any perturbation, rather than as violations of specific semantic invariants. Both methods detect the same underlying signal: world models prone to W2R failure produce inconsistent outputs whether perturbed by semantically meaningful state-preserving commands or by stochastic sampling noise. This finding implies that the discriminative power of our metamorphic audit does not derive from the domain-specific structure of the `look` command, but rather from its role as a generic perturbation that exposes model fragility.

Table 3: Per-seed breakdown of stutter-invariance vs. B3 sampling consistency. Bootstrap 95% CIs include zero on all seeds, and the direction of difference is inconsistent (B3 wins on seed 0).

| Seed | AUROC (Stutter) | AUROC (B3) | Delta | Bootstrap 95% CI |
|---|---|---|---|---|
| 0 | 0.732 | 0.748 | −0.015 | [−0.081, 0.052] |
| 1 | 0.767 | 0.740 | +0.027 | [−0.031, 0.092] |
| 2 | 0.801 | 0.784 | +0.017 | [−0.043, 0.075] |

**Practical Implications.** For practitioners deploying LLM-based world models, our results suggest that the cheapest stability check—simply re-running generation with different random seeds—is sufficient for W2R failure prediction. Domain-specific metamorphic probes add computational cost (3× context length for stutter insertion, embedding model inference for distance computation) without measurable benefit. When computational resources are limited, sampling consistency provides equivalent predictive power at lower cost.

**Limitations.** Our evaluation is limited to a single domain (TextWorld), a single world model architecture (Qwen2.5-7B), and a single acting agent (GPT-4o-mini). The equivalence between structured and generic stability testing may not hold in other settings where failure modes are more specific to particular state transitions. Additionally, our stutter-invariance method underwent post-hoc optimization while B3 did not, potentially biasing the comparison in favor of our approach. See Appendix A for implementation details.

**Future Work.** Several directions merit investigation: (1) other metamorphic relations such as action reordering or synonym substitution that may capture different failure modes; (2) evaluation on other text environments (ALFWorld, WebShop, ScienceWorld) where state-preserving commands have different semantics; (3) identifying conditions under which structured probes do provide advantage over generic stability testing.

## 6 Conclusion

We proposed a stutter-invariance metamorphic audit for predicting World-to-Real transfer failure in text world-model rollouts. By inserting state-preserving `look` commands and measuring observation drift, our method achieves AUROC 0.767 for W2R failure prediction. However, this performance is statistically tied with a simpler sampling consistency baseline (AUROC 0.757), indicating that domain-specific metamorphic probes do not provide measurable advantage over generic stability testing for this task. This informative negative result suggests that world model failures manifest as general output instability detectable by the cheapest form of perturbation—re-sampling with different random seeds.

## References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, D. Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, S. Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, O. Boiko, Made laine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, L. Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, C. Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, J. Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, S. Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, W. Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, B. Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, I. Kanitscheider, N. Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, J. Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Li, Rachel Lim, Molly Lin, Stephanie L. Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, A. Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, S. McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, An drey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, O. Murk, David M’ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O’Keefe, J. Pachocki, A. Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, J. Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, N. Ryder, M. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, M. Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, N. Staudacher, F. Such, Natalie Summers, I. Sutskever, Jie Tang, N. Tezak, Madeleine Thompson, P. Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer’on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. 2023.

Hyungjoo Chae, Namyoung Kim, Kai Tzu iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. *ArXiv*, abs/2410.13232, 2024.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. pp. 2318–2335, 2024.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, B. Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. pp. 41–75, 2018.

Yu Gu, Boyuan Zheng, Boyu Gou, Kai Zhang, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your llm secretly a world model of the internet? model-based planning for web agents. *ArXiv*, abs/2411.06559, 2024.

David R Ha and J. Schmidhuber. World models. *ArXiv*, abs/1803.10122, 2018.

Danijar Hafner, J. Pašukonis, Jimmy Ba, and T. Lillicrap. Mastering diverse domains through world models. *ArXiv*, abs/2301.04104, 2023.

Michael Janner, Justin Fu, Marvin Zhang, and S. Levine. When to trust your model: Model-based policy optimization. *ArXiv*, abs/1906.08253, 2019.

Rahul Kidambi, A. Rajeswaran, Praneeth Netrapalli, and T. Joachims. Morel : Model-based offline reinforcement learning. *ArXiv*, abs/2005.05951, 2020.

Jiapeng Li, Zheng Zheng, Yuning Xing, Daixu Ren, Steven Cho, and Valerio Terragni. Metamorphic testing of deep reinforcement learning agents with mdpmorph. *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 4086–4089, 2025a.

Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Pony Ma, Guanhua Chen, Heng Ji, and Mengdi Wang. From word to world: Can large language models be implicit text-based world models?, 2025b. URL `https://arxiv.org/abs/2512.18832`.

T. Moerland, J. Broekens, and C. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16:1–118, 2020.

Julian Schrittwieser, Ioannis Antonoglou, T. Hubert, K. Simonyan, L. Sifre, Simon Schmitt, A. Guez, Edward Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604 – 609, 2019.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *ArXiv*, abs/2010.03768, 2020.

Yuchi Tian, Kexin Pei, S. Jana, and Baishakhi Ray. *DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars*. 2017.

Ruoyao Wang, Peter Alexander Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? pp. 11279–11298, 2022.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Alexander Jansen. Can language models serve as text-based world simulators? pp. 1–17, 2024.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *ArXiv*, abs/2207.01206, 2022.

Tianhe Yu, G. Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, S. Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *ArXiv*, abs/2005.13239, 2020.

## A Implementation Details

Our implementation uses the Word2World evaluation framework with the following configuration. The world model (Qwen2.5-7B fine-tuned on TextWorld) is served via vLLM with a maximum context length of 65,536 tokens. For the stutter-invariance audit, we use deterministic decoding (temperature $= 0$, top-$p = 1.0$) to ensure reproducibility. The embedding model for drift measurement is BGE-M3, which produces 1024-dimensional embeddings. For the sampling consistency baseline (B3), we use stochastic decoding with temperature $= 0.7$ and top-$p = 0.9$.

The aggregation strategy for the stutter-invariance score was selected through a systematic search over 15 candidates, including mean, max, sum, and various weighted combinations. The exponentially-weighted sum ($\sum_t d_t \cdot \exp(t/T)$) achieved the best performance, improving AUROC from 0.713 (with simple mean aggregation) to 0.767. We note that this post-hoc optimization was not applied to the baselines, which may bias the comparison in favor of our method.