Outcome Is Not Enough: Trace-Based Lifecycle Evaluation for Trustworthy Agentic Economic Systems

Peiying Zhu
Blossom AI
San Francisco, USA

Sidi Chang
Blossom AI Labs
Tokyo, Japan

ABSTRACT

Agentic economic systems are often evaluated with scalar business outcomes: revenue, conversion value, spend efficiency, or latency. These signals are objective and scalable, but they can be incomplete when the agent acts under hidden state and changes the system it is evaluated within. We study this failure in a two-hotel pricing benchmark. A learning pricing agent can achieve plausible revenue per available room while failing to preserve the rate discipline of a rule-based revenue-management benchmark. We propose *discipline stability*, a trace-based lifecycle evaluation protocol: define the benchmark behavior, restrict observations to the deployable regime, induce trace diagnostics from failure, separate mechanisms with ablations, and test transfer, monitoring, and deployment after direct benchmark access is reduced. Across reward-only PPO variants, information-symmetric behavior-cloning baselines, hidden-state diagnostics, corrected-history student transfer, frozen deployment tests, and a compact hidden-budget bidding task, scalar outcomes alone miss trustworthiness failures that trace diagnostics expose. The contribution is an evaluation and monitoring framework for agentic economic systems with hidden state and evolving feedback loops, not a new optimizer or a universal claim about multi-agent reinforcement learning.

1 INTRODUCTION

Production agentic AI systems are increasingly asked to act, not only predict: they route requests, allocate budgets, choose prices, call tools, and adapt over multi-step interactions. In such settings, an outcome metric can be easy to log but hard to trust. A deployed agent may optimize a local KPI while violating the behavioral discipline that made the benchmark system reliable.

Recent agent benchmarks study tool use, web and desktop control, software tasks, interactive decision making, ML experimentation, and multi-turn evaluation [1, 2, 5, 6, 10, 11, 13]. We study a complementary production-evaluation question: when an economic agent acts in a feedback system, can the evaluator detect whether the agent preserves the behavioral structure of the benchmark, not only whether it achieves the headline score?

This paper studies that evaluation problem in strategic economic agents. The testbed is hotel pricing, but the central issue is broader: when an agent acts under partial observability, the same scalar outcome can be produced by several latent states and several valid behavioral traces. This is a form of proxy failure related to reward gaming [9] and reward misspecification [8]. It is also aligned with recent critiques of agent evaluation practice: accuracy-only benchmarks can hide cost, shortcutting, weak holdouts, and reproducibility failures [3]. Our setting replaces single-task success with business outcomes plus trace evidence. The practical evaluation question is:

> Can we certify not only that an agent achieves the target outcome, but that it preserves the benchmark behavior that produced that outcome?

We call this property *discipline stability*. A policy is discipline-stable relative to a benchmark if it preserves both the scalar outcome and the trace structure of the benchmark under the information regime in which the policy is deployed. This framing is useful for agentic AI evaluation because it turns post-deployment monitoring into a concrete trace question: which behavioral statistics should remain stable after training, transfer, model updates, or removal of privileged benchmark signals?

Our main benchmark is a two-hotel pricing simulator. Hotel A is the learner. Hotel B is a fixed revenue-management (RM) competitor using a deterministic rule based on time, market condition, and its own hidden remaining inventory. Guests choose Hotel A, Hotel B, or the outside option. Hotel A can observe its own state and, in competitor-aware regimes, lagged market prices; it does not observe Hotel B's inventory or pricing rule. Thus the environment is a single-agent partially observable control problem from Hotel A's perspective [7], although Hotel A's prices change which hotel receives demand.

The pricing benchmark and trace-prior setup are reused here as a compact testbed for lifecycle evaluation. The main object is not a hotel pricing optimizer, but an audit protocol for checking whether a deployed economic agent preserves benchmark discipline under hidden state.

The paper makes four KDD-relevant contributions. First, we define a trace-based evaluation protocol for agentic economic systems whose scalar KPIs are not enough for trustworthiness. Second, we show a reproducible failure: reward-only PPO variants can miss the benchmark trace even when the scalar objective is natural. Third, we diagnose the trigger: hidden competitor inventory makes one visible state compatible with several valid benchmark prices, so deterministic copying collapses uncertainty. Fourth, we test persistence: a corrected-history student can inherit much of the teacher's discipline, and frozen learned-agent matchups do not immediately collapse after the fixed benchmark is removed.

We keep the claim boundary explicit. We do not claim real hotel validity without calibration, do not claim that all modern MARL methods fail, and do not claim online co-learning safety. The contribution is a lifecycle evaluation and monitoring paradigm for agentic systems whose intended behavior is richer than a scalar KPI.

2 PRICING BENCHMARK AND HIDDEN STATE

Each pricing episode has horizon $H = 30$, default capacities $Q_A = Q_B = 100$, and price grid $\mathcal{A} = \{100, 120, 140, 160, 180, 200, 220\}$. At

Table 1: Information regimes. NC has no direct competitor-price context. CA-lag observes lagged market prices. Neither deployable regime observes current competitor inventory $q_{B,t}$ or the RM rule.

Regime	Observed by learner	Hidden from learner
NC	own state, own history	$q_{B,t}$, RM rule
CA-lag	NC + lagged market prices	current $q_{B,t}$, RM rule
Oracle	CA-lag + $q_{B,t}$	RM rule

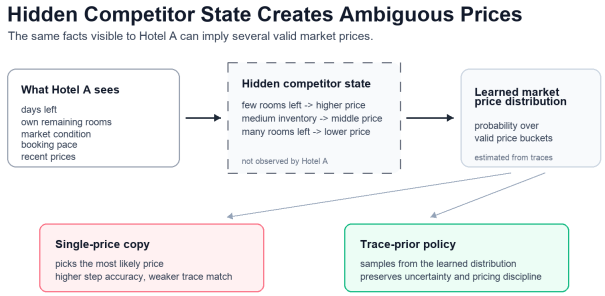

Figure 1: Hidden-state aliasing. The same visible state can imply multiple valid benchmark actions. Deterministic copying collapses the mixture; trace diagnostics preserve it.

step t, Hotel A chooses $p_{A,t}$ and Hotel B chooses

$$p_{B,t} = \text{FixedRM}(t, q_{B,t}, m_t),$$

where $q_{B,t}$ is Hotel B's remaining inventory and m_t is the market condition. The fixed RM rule is a hand-tuned OSG-style price rule plus a booking-pace floor. It is deterministic, but the learner cannot observe its private inputs.

Hotel A's scalar reward is per-step revenue normalized by starting capacity:

$$r^{\text{out}}_{A,t} = \frac{p_{A,t} y_{A,t}}{Q_A},$$

where $y_{A,t}$ is the number of rooms sold by Hotel A at step t. Summing this quantity gives the simulator's RevPAR measure. This scalar reward naturally encourages selling rooms, but it does not distinguish disciplined yield management from low-price occupancy grabbing.

The POMDP trigger is hidden $q_{B,t}$. The same Hotel A-visible observation can correspond to several competitor inventories and therefore several valid benchmark prices:

$$\pi_M(a \mid o_t) \approx \Pr(a_{B,t} = a \mid o_t).$$

The right target is a posterior predictive distribution, not a single action label.

3 DISCIPLINE STABILITY

Let $\pi^\star$ be a benchmark policy or trace reference, ρ an information regime, C a set of state slices, and $\mathcal{Z}$ trace variables such as price, sales, occupancy, ADR, remaining inventory, and pacing behavior. A policy π passes a discipline-stability check at tolerances $(\epsilon_{\text{out}}, \epsilon_z)$ if

$$\Delta_{\text{out}}(\pi, \pi^\star; \rho) \le \epsilon_{\text{out}}, \tag{1}$$

$$D(P^z_\pi(\cdot \mid c, \rho), P^z_{\pi^\star}(\cdot \mid c, \rho)) \le \epsilon_z, \quad \forall z \in \mathcal{Z},\ c \in C. \tag{2}$$

Here Δ_{out} is the absolute expected outcome gap. For price and bid distributions we report $D_1(P, Q) = \sum_x |P(x) - Q(x)|$ and Jensen-Shannon divergence.

The protocol has five steps:

(1) define the benchmark discipline beyond the KPI;
(2) specify the deployable observation regime and hidden variables;
(3) induce trace diagnostics from the observed failure;
(4) run ablations that separate weak optimization, hidden state, deterministic collapse, and distributional repair;
(5) test whether the discipline persists after benchmark access is reduced or removed.

This is designed as a lifecycle evaluation tool. Pre-deployment tests can reveal whether the scalar KPI is achieved for the wrong reason. Post-market monitoring can track whether trace distributions drift after model updates, context changes, or upstream policy changes. Governance reviews can use the same traces to ask what the agent did, not only what score it achieved. In production terms, the evaluator should preserve the raw event log needed to recompute each trace statistic after a model or policy update: observed state slices, action buckets, realized outcomes, benchmark or reference actions when available, and per-seed or per-period aggregation metadata.

4 NEGATIVE CONTROLS AND TRACE-ACCESS BASELINES

We first ask whether stronger reward-only RL is enough. PPO-style baselines, including recurrent and centralized-training/decentralized-execution variants, are relevant because recurrent model-free policies can be strong under partial observability [7], and PPO variants are strong baselines in multi-agent settings [12]. All three optimize only $r^{\text{out}}_{A,t}$. In CTDE PPO, the critic receives hidden competitor information during training, while the deployed actor remains decentralized.

Reward-only baselines are negative controls, not information-symmetric competitors. We therefore also run trace-access baselines: behavior cloning (BC)-only stochastic copy, BC warm-start plus PPO, and PPO with a sampled-action BC auxiliary. Table 3 shows the important nuance. In the default symmetric market, BC-only stochastic copy is nearly tied with the Trace-Prior teacher. This means the trace prior itself carries much of the discipline. The algorithmic claim should therefore stay modest: Trace-Prior RL is most useful when the agent must preserve a benchmark trace while optimizing a local objective that is not exactly "copy the benchmark."

Under capacity asymmetry, the RL component becomes more visible. With $Q_A = 120$ and $Q_B = 100$, exact copying is conservative for the larger hotel. Trace-Prior RL improves paired RevPAR by +0.764 with 95% CI [+0.125, +1.403] and improves occupancy by +0.0051 with 95% CI [+0.0014, +0.0088]. The gain appears through filling more rooms at similar rates, not through charging higher

Table 2: Reward-only PPO variants fail the benchmark trace, while trace-aligned methods recover it. R-PPO denotes recurrent PPO.

Method	RevPAR A	RevPAR B	Occ A	Occ B	ADR A	ADR B	D_1^p	D_{JS}^p
PPO	93.554	106.925	0.7342	0.7964	127.68	134.21	0.4635	0.0559
R-PPO	95.026	101.202	0.7787	0.7677	122.30	131.76	0.5400	0.0657
CTDE PPO	94.482	110.182	0.7377	0.8000	128.26	137.71	0.6582	0.0959
Trace-Prior teacher	108.063	107.931	0.7705	0.7677	140.26	140.59	0.0165	0.0000
Corrected-history student	107.588	108.683	0.7670	0.7726	140.29	140.68	0.0198	0.0001

Table 3: Information-symmetric trace baselines. Trace access alone is powerful; full-distribution trace regularization is the repair used by the teacher.

Method	RevPAR A	D_1^p	D_{JS}^p
Reward-only PPO	93.554	0.4635	0.0559
BC-only stochastic copy	107.931	0.0147	0.0000
BC warm-start + PPO	102.214	0.2054	0.0265
PPO + BC aux, $\alpha = 0.1$	108.093	0.0816	0.0059
PPO + BC aux, $\alpha = 1.0$	107.749	0.0477	0.0015
Trace-Prior teacher	108.063	0.0165	0.0000

rates. This is useful evidence for bounded adaptation, but not proof of broad algorithmic novelty.

5 MECHANISM: HIDDEN STATE AND DISTRIBUTIONAL REPAIR

We test the hidden-state mechanism directly. A market-price predictor using only deployable CA-lag features obtains NLL 0.5359 and exact price accuracy 76.91%. Adding oracle $q_{B,t}/Q_B$ improves NLL to 0.1557 and accuracy to 95.47%. This is not a deployable model; it is a diagnostic showing that hidden competitor inventory explains much of the label uncertainty.

The same mechanism explains why deterministic copying can fail. An argmax predictor can have higher step accuracy while distorting the aggregate trace: it collapses the posterior predictive mixture onto the modal price bucket. In the pricing benchmark, argmax prediction reaches 78.14% exact step accuracy but has worse aggregate alignment than stochastic sampling; sampling lowers step accuracy to 69.50% but improves D_1^p from 0.0323 to 0.0183. For lifecycle evaluation, this is the key lesson: action accuracy and trace alignment can move in opposite directions.

The Trace-Prior teacher trains a supervised prior $\hat{\pi}_M$ from observed market traces and then optimizes

$$J(\theta) = \mathbb{E}_{\xi\sim\pi_\theta}\left[\sum_{t=0}^{H-1}\left(r_{A,t}^{\mathrm{out}} - \beta\, D_{\mathrm{KL}}(\pi_\theta(\cdot \mid o_t) \,\|\, \hat{\pi}_M(\cdot \mid o_t))\right)\right]. \quad (3)$$

The coefficient β interpolates between unconstrained outcome optimization and trace discipline. We use $\beta = 30$ for the selected teacher. This is related to recent offline-to-online RL [4], but the prior here estimates a benchmark trace under hidden state rather than the agent's own offline behavior.

Table 4: Selected student validation. The teacher metric lies inside the student's 95% CI for the main business metrics.

Metric	Student mean [95% CI]	Teacher	Inside CI?
RevPAR	107.960 [107.026, 108.894]	108.060	yes
Occupancy	0.7728 [0.7604, 0.7852]	0.7703	yes
ADR	139.71 [138.46, 140.95]	140.29	yes

6 PERSISTENCE AND DEPLOYMENT DIAGNOSTICS

A post-market evaluation protocol must ask whether discipline persists after privileged benchmark access is reduced. We therefore train a corrected-history student using a dataset-aggregation-style correction idea. The student uses Hotel A-visible state and its own recent corrected price history:

$$o_{S,t} = (\tau_t/H, q_{A,t}/Q_A, m_t, h_{A,t}, a_{A,t-1}, a_{A,t-2}, a_{A,t-3}).$$

Missing early lags are padded with a fixed start-of-episode token outside the price grid. The teacher labels states induced by the student's own rollouts. At deployment, the teacher is removed.

We then freeze learned policies and remove the Fixed RM benchmark. This is not online co-learning; it is a deployment diagnostic. NC-vs-NC and CA-vs-CA matchups remain symmetric. In seat-balanced mixed deployment, CA has a small directional advantage: RevPAR 107.891 versus 107.471, occupancy 0.7738 versus 0.7711, and ADR 139.44 versus 139.37. The supported claim is modest but important for trustworthiness monitoring: learned policies do not immediately collapse back to the low-price shortcut after the benchmark is removed.

7 SECOND DOMAIN: HIDDEN-BUDGET BIDDING

To test whether the paradigm is hotel-specific, we add a compact hidden-budget bidding task. A bidding policy chooses bid levels for sequential ad opportunities. The full-state expert observes remaining campaign budget and paces spend. Partial policies see time, public quality, competition, and lagged bid, but not hidden budget. The scalar outcome is conversion value per step; the discipline trace is bid distribution plus pacing behavior.

Across four variants with 10 seeds each, the full-state expert obtains value/step 0.482 and pacing gap 0.043. Outcome-only aggressive bidding is worse on both outcome and discipline: value/step 0.374, pacing gap 0.212, $D_1 = 0.9859$, and $D_{\mathrm{JS}} = 0.1905$. Partial argmax imitation almost recovers scalar value (0.475) but remains

trace-wrong ($D_1 = 0.3549$, $D_{\mathrm{JS}} = 0.0394$). Partial trace-prior sampling gives slightly lower value (0.467) but best preserves discipline: pacing gap 0.059, $D_1 = 0.0130$, and $D_{\mathrm{JS}} \approx 0$.

This second domain supports the mechanism story: scalar outcome can remain close while discipline fails; deterministic argmax can improve outcome while collapsing hidden-state uncertainty; stochastic trace-prior sampling preserves the expert trace.

8 GOVERNANCE, MONITORING, AND LIMITATIONS

For agentic AI evaluation, the practical output is not only a training method. It is a monitoring stack:

- scalar KPI: revenue, conversion value, latency, or resource use;
- business decomposition: occupancy and ADR, or spend and pacing;
- trace distribution: price or bid buckets, D_1, D_{JS};
- state slices: early/late horizon, high/low inventory, tight/loose budget;
- persistence tests: student transfer, frozen deployment, and mixed information-regime matchups.

These logs are useful for production governance because they describe how the agent achieved its score. They can support drift detection after model updates, detect shortcut optimization, and make audit reviews more concrete. In pricing or bidding systems, the question is not merely whether the KPI improved; it is whether the behavioral trace remains compatible with the benchmark discipline.

The limitations are substantial. The simulator is synthetic and not calibrated to real hotel data. Hotel B is a fixed benchmark, not a learning competitor. The bidding task is compact. Trace priors are only as good as the benchmark trace; if the trace is noisy, biased, collusive, or commercially undesirable, the method can preserve the wrong discipline. MAPPO [12], MADDPG-style online co-learning, real-data calibration, and model-update drift tests are natural next steps.

9 CONCLUSION

Outcome is not enough for trustworthy agentic economic systems. A policy can hit a scalar KPI while failing to preserve the behavior that makes the policy deployable. Discipline stability turns this into an evaluation problem that can be measured: define the benchmark discipline, restrict the observation regime, induce trace diagnostics, separate mechanisms with ablations, and test whether the behavior persists after direct benchmark access is reduced. The result is a concrete lifecycle evaluation protocol for agentic systems whose failures live in the trace, not only in the final score.

A REPRODUCIBILITY NOTES

The workshop submission is intended to be reproducible from the accompanying repository. All experiments use simulator-generated data, fixed seed lists, and per-seed evaluation summaries before aggregation. No private hotel or advertising data are used.

Code layout and environment. The benchmark, training scripts, and analysis utilities live in the hotel-pricing project directory. The project is pure Python and declares its numerical and deep-learning dependencies in the project configuration. From that directory, create an environment with Python 3.12 or newer and install the package in editable mode with the development extras.

Minimal command sequence. From the repository root:

(1) `cd projects/hotel-pricing-marl`
(2) `python -m venv .venv` and activate it.
(3) `python -m pip install -e ".[dev]"`
(4) Run the release commands in Table 5. Raw run outputs can be kept separate; for the reproducible release, organize the generated JSON, Markdown, and model files under the output folders shown in the table.

Run settings. The main PPO rows use CA-lag3 observations and three policy modes: feed-forward PPO, recurrent PPO, and CTDE PPO. Each uses seeds 0–9, 2,000 training episodes, and 1,000 evaluation episodes per seed. The selected Trace-Prior teacher uses seeds 0–9, $\beta = 30$, 2,000 training episodes, 10,000 evaluation episodes per seed, entropy coefficient 0, and prior temperature 0.95. The selected student uses hard teacher labels, history lag 3, two student-rollout correction rounds, no final RL polish, seeds 0–4, and 10,000 evaluation episodes per seed. The bidding check uses four variants: default, tight-budget, high-competition, and volatile-quality. It uses seeds 0–9, 2,500 training episodes, 1,200 evaluation episodes, and sampling temperature 0.95.

Aggregation. After raw runs finish, collect seed-level JSON summaries into `results/tables/`. Then run the table builder listed in Table 5 to combine those summaries into the table values reported in the paper. This aggregation step does not rerun training or evaluation. All reported confidence intervals are computed across seed-level means using the same aggregation helpers as the report scripts.

REFERENCES

[1] Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. arXiv:2310.03302 [cs.LG] https://arxiv.org/abs/2310.03302
[2] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. In *International Conference on Learning Representations*. https://arxiv.org/abs/2310.06770
[3] Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. 2024. AI Agents That Matter. arXiv:2407.01502 [cs.LG] https://arxiv.org/abs/2407.01502
[4] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=68n2s9ZJWF8
[5] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*. https://openreview.net/forum?id=zAdUB0aCTQ
[6] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. In *Advances in Neural Information Processing Systems*, Vol. 37. https://proceedings.neurips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html Datasets and Benchmarks Track.
[7] Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. 2022. Recurrent Model-Free RL Can Be a Strong Baseline for Many POMDPs. In *Proceedings of*

Table 5: Files and commands for reproducing the reported results. Commands are under `scripts/kdd2026/`; output folders are under `results/`.

Paper claim	Release command	Output folder
Reward-only PPO variants	`pricing_reward_only_ppo.py`	`pricing/reward_only_ppo/`
Trace-Prior teacher	`pricing_trace_prior_teacher.py`	`pricing/trace_prior_teacher/`
Argmax vs. sampling copy	`pricing_market_copy.py`	`pricing/market_copy/`
Corrected-history student	`pricing_student_train.py`; `pricing_student_validate.py`	`pricing/corrected_history_student/`
Frozen deployment diagnostics	`pricing_frozen_deployment.py`	`pricing/frozen_deployment/`
Hidden-budget bidding	`bidding_hidden_budget.py`	`bidding/hidden_budget/`

the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162). PMLR, 16691–16723. https://proceedings.mlr.press/v162/ni22a.html

[8] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=JYtwGwIL7ye

[9] Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and Characterizing Reward Gaming. In *Advances in Neural Information Processing Systems*, Vol. 35. https://proceedings.neurips.cc/paper_files/paper/2022/hash/3d719fee332caa23d5038b8a90e81796-Abstract-Conference.html

[10] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv:2404.07972 [cs.AI] https://arxiv.org/abs/2404.07972

[11] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045 [cs.AI] https://arxiv.org/abs/2406.12045

[12] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems*, Vol. 35. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9c1535a02f0ce079433344e14d910597-Abstract-Datasets_and_Benchmarks.html Datasets and Benchmarks Track.

[13] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv:2307.13854 [cs.AI] https://arxiv.org/abs/2307.13854