Offline Preference-Based Trajectory Evaluation

Fernando Diaz
Carnegie Mellon University
Pittsburgh, PA
diazf@acm.org

Abstract

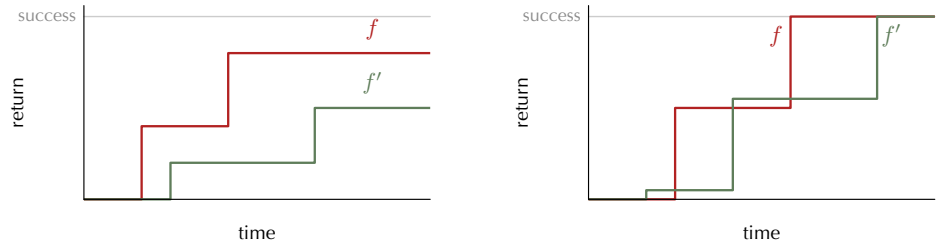
Offline evaluation of agentic systems often collapses trajectories to terminal success, discarding information about partial progress and inducing widespread ties, creating substantial statistical inefficiency by reducing effective sample size and weakening the ability to distinguish systems. We propose preference-based trajectory evaluation, which compares trajectories directly through temporal preferences over progress and time-to-return profiles. We find that, across diverse agentic and interactive benchmarks, standard success-based metrics produce tied comparisons on roughly 75% of instances, whereas trajectory-aware preferences reduce ties to roughly 35%, improving discriminative power, ranking stability, and data efficiency. Our results suggest that benchmark saturation, often portrayed as the result of poor data collection or problem difficulty, may also be explained by the choice of evaluation measure.

1 Introduction

Motivated by the increasing cost of evaluating AI systems [13], we study two core desiderata for evaluation metrics: sensitivity and data efficiency. Originally proposed by Mandel and Stiehler [25], sensitivity, in the context of AI evaluation, refers to a metric's ability to detect meaningful performance differences between systems under a fixed evaluation budget and becomes important as the performance of AI systems improves to a quality where even substantively different behaviors can result in small observed differences. Data efficiency is related to statistical power [8] and refers to the number of evaluation samples required to reach reliable comparative assessment of systems. Although distinct, these two objectives are related since insensitive metrics require substantially larger evaluation sets to resolve the same system differences that more sensitive metrics can detect with fewer observations. Framed this way, evaluations need to be both valid and statistically robust under real-world constraints.

Unfortunately, popular evaluation approaches adopt relatively insensitive and inefficient metrics that answer the evaluation question, 'did the system ever solve the task instance?' While convenient, in agentic systems, basing a metric on binary success measurements poses two problems. First, binary measurement often collapses partial solutions to 0, losing granular evaluation signals and conflating trajectories that may differ in progress toward a solution (Figure 1a). Second, binary measurement collapses the performance accrued over multi-step trajectories into a single scalar value, comparing two trajectories using their terminal values instead of how performance develops over time (Figure 1b). Combined, these two issues can compromise sensitive and efficient evaluation. To understand how, we can look at the number of ties between systems when using success as a measure since a large number of ties degrades the effective evaluation set size. In the benchmarks we study, an average of 75% of instance-level comparisons are ties under success rate. Even when comparing partial returns, the tie rate remains high at 50%. As systems become more performant, this inefficiency compounds because more trajectories are collapsed as indistinguishably successful, prompting claims of benchmark saturation [32, 42, 18]. At the same time, the use of binary success

Preprint.



(a) $\mu_{\mathrm{SR}}(f) = \mu_{\mathrm{SR}}(f') = 0$ (b) $\mu_{\mathrm{SR}}(f) = \mu_{\mathrm{SR}}(f') = 1$

Figure 1: **Trajectory ties under success rate.** (a) Two unsuccessful trajectories can be distinguished by partial returns. (b) Two successful trajectories can be distinguished by how they accumulate return over time.

metrics across machine learning and natural language processing conferences is increasing. We found that the percentage of abstracts in papers published at the NeurIPS Datasets and Benchmarks track mentioning binary metrics rose from 5% in 2022 to 18% in 2025; at EMNLP, the fraction rose from 9% in 2022 to 21% in 2025 (details in Appendix A). Together, these observations suggest that the current use of success rate is both inefficient and growing in adoption.

While many existing approaches to address evaluation inefficiency focus on reducing the number of test instances while maintaining success rate as the metric, we approach insensitivity and inefficiency by interrogating the systematization of 'performance' itself. Instead of collapsing each trajectory to a binary terminal value, we expand performance to capture how return progresses over time. This allows us to compare trajectories by adopting the principle of temporal preference: given two systems achieving the same task progress, we prefer the system that reached it sooner. Inspired by recent results in information retrieval evaluation [27, 10], we operationalize this principle in a family of measures that directly capture the preference without intermediary scalar metric computation. Importantly, our approach makes few assumptions beyond temporal preference and requires no additional hyperparameters, unlike methods based on temporal discounting.

We assess our proposed methods across multiple benchmark families spanning both classic reinforcement learning environments and contemporary agentic tasks. Our results show that trajectory-aware preferences reduce tie rates from roughly 75% to 35% on average, recovering a substantial portion of previously discarded signal. By preserving information, our methods lead to consistent improvements across standard measurement criteria: higher reliability, sensitivity, and data efficiency. More broadly, our results suggest that benchmark saturation can result from the information loss induced by metric definition. While the benefits of our approach may appear intuitive, current benchmarks adopt success rate, and the statistical consequences of this design choice have not been systematically studied.

2 Background

Modern evaluations face two increasingly visible limitations. First, as models improve, many established benchmarks show signs of saturation and can no longer reliably distinguish among high-performing systems [32, 42, 18, 2]. Second, large-scale evaluation has become prohibitively expensive. Ghosh et al. [13] show that evaluating a single system on a modern benchmark can require several thousand dollars in inference costs alone. Taken together, these trends suggest that the benchmarks that are most expensive to run are often those least able to resolve meaningful differences between models.

As a result of these limitations, there have been increasing calls for more principled and rigorous evaluation [31]. Approaches can be roughly divided into two categories. The first category fixes the evaluation metric and develops methods to sample or weight instances to improve efficiency. Methods include dynamic benchmarking [22, 37], robust statistical practices [1, 14, 7], active learning approaches [19, 15, 4, 20, 28], and item response theory [26, 42, 41, 34, 23, 30]. The second category focuses on the development of improved measurement instruments by adopting methods from measurement theory to design metrics [44, 7], providing a theoretical framework to inspect the systematization of a concept (i.e., the relevant factors considered when measuring the concept) and its operationalization (i.e., how we detect and quantify the relevant factors). In the case of success rate,

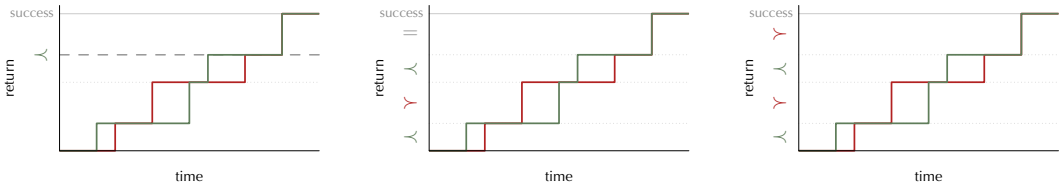
	A		B		
task	success	time	success	time	winner
1	1	2	1	1	B
2	1	11	1	10	B
3	1	11	1	10	B
4	1	11	1	10	B
5	1	1	0	-	A
mean	1	7.2	0.8	7.75	-

Table 1: **Aggregation reversal in decoupled success-rate and time-to-success evaluation.** Example success rate and conditional mean time-to-success for two models over five tasks. The last column reflects which model has a faster time-to-success for the task. Separately computing success rate and mean time-to-success erroneously suggests that A dominates B across both metrics when in fact B dominates A in 80% of tasks.

'performance' may be systematized as 'whether the agent completed the task' and operationalized as 'whether the agent is in a pre-defined end state.' Recent calls to consider system cost during evaluation [17] can be interpreted as expanding the systematization of performance to include inference cost, which is then operationalized as 'distance from the Pareto frontier of cost and task completion.'

Among dimensions that contribute to performance, time plays an important role for agentic systems. Time efficiency has long been an important factor in system performance, changing the question from 'can the system complete the task?' to 'can the system complete the task in a reasonable amount of time?' This is natural in the evaluation of agents since they complete tasks by interacting with the environment over multiple steps. While many benchmarks calculate the average number of steps in addition to success rate, inspecting these metrics independently can conceal task instances where two systems succeed but with dramatically different time efficiency, resulting in system order reversals (Table 1). The temporal choice literature distinguishes between temporal preference—an ordering over outcomes with identical utility occurring at different times—and temporal discounting—a particular modulation of an outcome's utility based on when it occurs in time [12]. In the context of comparing two trajectories, temporal preference would specify that, if both trajectories succeed, prefer the shorter trajectory. Temporal discounting, by contrast, converts time into a scalar weight applied to performance, reducing the ordering of trajectories to comparing discounted utilities. In reinforcement learning, linear discounting (subtracting a constant penalty per step) and exponential discounting (rescaling future rewards multiplicatively by a constant factor) are often introduced for algorithmic convenience in optimization rather than as a principled evaluative criterion, which is most often the undiscounted return [38]. When time is considered, researchers often use power-law discounting, reflected in the 'Success weighted by Path Length' metric [3], which divides binary success by the ratio of trajectory length over the optimal path length. That said, in real world settings, temporal discount rates can vary by domain and are non-stationary [12], making these methods brittle since they assume a precise relationship between time and utility. To avoid these issues, we adopt evaluation methods based on temporal preference which are based on fewer assumptions and do not require additional hyperparameters.

Temporal preference is part of a broader class of approaches that shift from assigning scalar values to model outputs or behaviors (metric-based evaluation) to assigning signed values to pairs of model outputs (preference-based evaluation). Although evaluation based on paired comparisons is an established method for variance reduction and improved data efficiency [33], these methods have been largely absent from offline machine learning evaluation. When paired comparisons arise, it is normally through online preference-based evaluation [6, 16], where explicit or implicit feedback from human users is used to assess which of two models' outputs is preferred. When using online methods, comparing a new model requires collection of new data, which can be prohibitive during model development, due to experimentation speed and safety requirements. Our work can be seen as the offline counterpart of online arena-style preference-based evaluation. As such, it inherits the benefits of offline evaluation, including counter-factual analysis (i.e., comparing more than two systems in the same context), safety, and speed.



(a) Lexicographic Return (b) Return-Paired Preference (c) Interval-Paired Preference

Figure 2: **Trajectory preferences based on time to return.** Evaluation operates by comparing pairs of trajectories and aggregating preferences within trajectories. (a) Lexicographic Return (LR) derives a preference from the time to reach the earliest non-tied return. (b) Return-Paired Preference (RPP) integrates time-to-return across all return levels. (c) Interval-Paired Preference (IPP) compares the time between return levels or sub-goals.

3 Preference-Based Trajectory Evaluation

We are interested in broadening the systematization of performance beyond binary success to include richer trajectory information.

We represent a trajectory as a function $f : \mathbb{Z}^+ \to [0, 1]$, where $f(t)$ is the normalized return at discrete time step t and a return of 1 indicates task success. We consider domains where incremental rewards are non-negative and, as a result, $f(t)$ is nondecreasing in t. The *time-to-return* is a function $g : [0, 1] \to \mathbb{Z}^+ \cup \{\infty\}$, where $g(R)$ is the first time at which the agent achieves a return of at least R; if R is never reached, $g(R) = \infty$.

An *evaluation metric* is defined as $\mu(f) \in [0, 1]$. At an instance level, success rate (SR) systematizes performance as (binary) task completion, $\mu_{\text{SR}}(f) = \text{I}[g(1) < \infty]$; time and partial progress are both excluded. Partial return (PR) systematizes performance as the progress made toward task completion, $\mu_{\text{PR}}(f) = \max_t(f(t))$; partial progress is included but time is still excluded. The most common evaluation metric that considers both time and task completion is the power law discounted success rate (SPL), $\mu_{\text{SPL}}(f, k) = g(1)^{-k}$; this assumes a power law relationship between time and utility.

Recent work in information retrieval has introduced preference-based measures as more sensitive, metric-free methods for evaluation [27, 10]. Given two trajectories f and f', we define an *evaluation preference* as $\Delta(f, f') \in [-1, 1]$, where $\Delta > 0$ indicates that f is preferred, $\Delta < 0$ indicates f' is preferred, and $\Delta = 0$ indicates indifference. An evaluation metric can be represented as a preference by computing $\Delta(f, f') = \mu(f) - \mu(f')$.

Although we can derive a preference from a metric, we can also directly design evaluation preferences that consider alternative systematizations. In what follows, we will introduce several evaluation preferences that capture composite success and temporal preference systematizations and, as a result, do not require a precise relationship between measured time and success.

Lexicographic Return (LR) The most conservative way to design a temporal preference reproduces $\Delta_{\mu_{\text{SR}}}$ when it is not zero and otherwise breaks ties based on time-to-return. The lexicographic return preference (LR) follows this logic by comparing two trajectories f and f' starting at the highest return level $R = 1$ (i.e. 'success'). If f is successful (i.e., $g(R) < \infty$) while f' is not (i.e., $g'(R) = \infty$), we say $f \succ f'$. If $g(R) = g'(R)$, we back off to the highest return level one model reaches before the other,

$$\Delta_{\text{LR}} = \text{sgn}\left[g'(R^*) - g(R^*)\right] \tag{1}$$

where $R^* = \max\{R \in [0, 1] : g(R) \neq g'(R)\}$. If no such R^* exists, $\Delta_{\text{LR}} = 0$. Figure 2a provides an example of computing LR for two trajectories. LR systematizes performance as relative temporal priority at the highest return difference. LR can be adopted in cases where a conservative alignment with success rate is desired, with ties broken by time or, if those are tied, lower return levels. LR is equivalent to 'lexicographic recall' in the information retrieval literature [11].

Return-Paired Preference (RPP) While LR looks at a single point where two trajectories differ, return-paired preference (RPP) sweeps uniformly across all return levels and, at each level, compares the time-to-return for both trajectories. Let $[\hat{R}_0, \ldots, \hat{R}_K]$ denote the sorted union of all return levels

achieved by either trajectory, augmented with 0 and 1. For each return segment $[\hat{R}_{k-1}, \hat{R}_k)$, if $g(\hat{R}_k) < g'(\hat{R}_k)$, then $f \succ f'$ for that segment. We then average those segment-level preferences, weighted by the segment width $\hat{R}_k - \hat{R}_{k-1}$:

$$\Delta_{\text{RPP}} = \sum_{k=1}^{K} (\hat{R}_k - \hat{R}_{k-1}) \operatorname{sgn}\left[g'(\hat{R}_k) - g(\hat{R}_k)\right]. \tag{2}$$

where $\infty - \infty = 0$. Figure 2b provides an example of computing RPP for two trajectories. RPP systematizes performance as cumulative temporal advantage across all return levels, assuming a uniform weighting across levels. As such, RPP is appropriate when we care about cumulative reward but are indifferent between return levels. This naturally emerges in many information seeking tasks and, as a result, the ranking analogue from information retrieval is 'recall-paired preference' [10].

Interval-Paired Preference Rather than comparing the absolute time-to-return, the interval-paired preference compares the *time increment* required to advance from one return level to the next. This arises when rewards are accumulated as sub-goals are reached. For each segment $[\hat{R}_{k-1}, \hat{R}_k)$, let $\delta_g^k = g(\hat{R}_k) - g(\hat{R}_{k-1})$ denote the additional time trajectory g requires to go from return $\hat{R}_{k-1}$ to $\hat{R}_k$, and define $\delta_{g'}^k$ analogously. Then,

$$\Delta_{\text{IPP}} = \sum_{k=1}^{K} (\hat{R}_k - \hat{R}_{k-1}) \cdot \operatorname{sgn}\left[\delta_{g'}^k - \delta_g^k\right]. \tag{3}$$

Figure 2c provides an example of computing IPP for two trajectories. IPP systematizes performance as local temporal efficiency at each incremental step, asking 'at each sub-goal transition, which system was faster to advance?' A trajectory that starts slowly but then makes faster local progress can be preferred under IPP even if it is not preferred under RPP, because IPP compares incremental transition times whereas RPP compares absolute time-to-return. The systematized concept is closer to consistency of progress than overall speed.

We note that, when intermediate rewards are missing (e.g., only success and number of steps are recorded), the three preferences are identical, by design.

4 Methods and Materials

We compare our preference-based methods—LR, RPP, and IPP—with metric-based methods—SR, PR, and SPL—when evaluating runs across five benchmarks, each of which contains one to six tasks, which, in turn, contain 30-500 task instances. Our goal is to understand the relative strengths of each, with respect to meta-evaluation desiderata (Section 4.3). We distinguish between task instance analysis, which looks at preferences between pairs of outputs conditioned on a specific task instance description; system pair analysis, which looks at preferences between pairs of systems across all task instances; and system ranking analysis, which looks at the ranking of systems across all tasks derived from a specific measurement approach.

4.1 Data

We evaluate across five benchmark families spanning interactive text, workspace, and software engineering settings. Each benchmark dataset contains trajectories for 12-54 models, run across all task instances. A summary of datasets can be found in Table 4 of Appendix B. AgentBoard (AB) [21] provides six task suites—ALFWorld, ScienceWorld, BabyAI, PDDL, WebShop, and Tool-Query—covering heterogeneous interactive environments with partial-progress subgoal scores across 12 evaluated systems. OpenHands-Index (OHI) [40] aggregates up to 22 systems across four code-generation benchmarks (SWE-bench, SWT-bench, SWE-bench-Multimodal, and GAIA). OHI only records terminal binary success as well as the number of steps, allowing us to demonstrate the efficacy of preference-based evaluation for trajectories lacking intermediate or partial rewards. TheAgentCompany (TAC) [45] is a suite of 175 workplace agent tasks. While TAC records partial terminal returns and step counts, it does not include intermediate rewards. Text Adventure Learning Environment Suite (TALES) [9] collects trajectories from roughly 50 systems across two

text-adventure environments (Jericho and ScienceWorld). We also generated the sub-goal reinforcement learning (SGRL) dataset of trajectories derived from agents acting in a variety of traditional reinforcement learning environments (DoorKey, FourRooms, Taxi) where progress can be measured by completing sub-goals.

In addition to these datasets, we assembled two auxiliary datasets to assess specific measure properties. In order to test whether a measure detects a difference between models when none exists, we include a same-model variant of TALES (TALES-AA) where we use two random trajectories from the same model for each task, treating them as having come from different models; this provides null-hypothesis data for measuring false-positive rates. We use the same tasks as SGRL to define a separate 'oracle' dataset (SGRL-oracle), where we have designed an optimal policy for the environment and introduced progressively more noise to provide a ground truth ordering of models (i.e., an optimal model with more interpolated noise will be inferior to a model with less noise).

Full details of our datasets are provided in Appendix B.

4.2 Evaluation Measures

As baselines, we consider success rate (SR) as well the (partial) terminal return (PR). We adopt a reference-free version of Success weighted by Path Length (SPL) [3], dividing the success indicator by the observed trajectory length.

We measure the preference between two systems by averaging the paired preference between model outputs across the set of task instances, $\frac{1}{|X|}\sum_{x\in X}\Delta(f_x, f'_x)$ where X is the set of task instances.

In order to generate a ranking of systems from pairwise preferences (see [43] for a survey), we adopt the Bradley-Terry model used in ChatBot Arena [6]. To do so, we extend the Bradley-Terry model to fractional labels, mapping each preference $\Delta(f_x, f'_x)$ to a soft win fraction $(\Delta(f_x, f'_x)+1)/2 \in [0, 1]$. We then fit a standard Bradley-Terry model by maximizing the cross-entropy likelihood. We leave the exploration of alternative aggregation methods to future work.

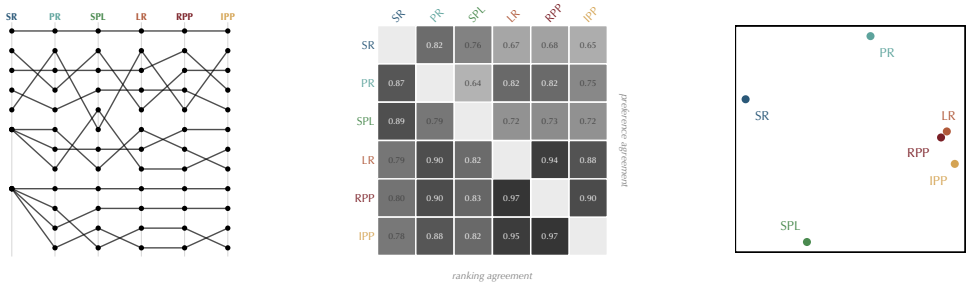
4.3 Meta-Evaluation

In order to compare evaluation measures, we adopt several criteria from measurement theory and metric design.

Validity ensures the measure actually captures the intended construct (e.g., system performance) without being redundant with existing measures of the same construct or misaligned with ground truth ordering. *Inter-measure similarity* provides us with a data-driven understanding of the relationship between measures. In this case, we are interested in a measure (a) being related to success rate, since both operationalize performance (i.e., convergent validity) but (b) not being so similar as to be redundant. We measure similarity at both the instance-level (pairwise agreement between measures) as well as ranking level (Kendall's τ correlation between system rankings derived from measures). *Oracle agreement* measures the number of true system preferences recovered by the measure; we use the SGRL-oracle dataset for these experiments. Oracle agreement uses both sign agreement as well as the number of statistically significant preferences detected, corrected for multiple comparisons (see 'Discriminative power' below).

Reliability ensures that a measure yields stable and consistent results under resampling or small perturbations of the data. In *split-half reliability*, we randomly partition the task instances into two equal halves, compute the mean metric value per model pair on each half, and measure the Kendall τ correlation between the two half-rankings. We repeat this for 100 random splits and report the mean correlation. Higher values indicate that the metric's system-level ranking is stable under subsampling. In *leave-one-out stability*, for each task instance, we remove it and recompute the sign of the mean difference for every model pair. We report the fraction of pairs for which removing any single instance causes a sign flip in the aggregate preference. A metric with few sign flips is robust to individual outlier instances.

Sensitivity ensures the measure can detect meaningful differences between systems. Unlike validity measures, we are only interested in detecting a difference, not detecting an accurate difference. *Tie rate* is an instance-level metric that computes the number of paired comparisons that result in a tie. *Discriminative power* computes the number of model pairs for which a metric detects a statistically



(a) Bump Chart (b) Inter-measure correlation (c) t-SNE embedding

Figure 3: **Inter-metric similarity:** (a) Bump chart showing the ranking of systems across several measures for AgentBoard ALFWorld runs; charts for other tasks can be found in Figure 6. (b) Correlation between measures. Upper triangle: agreement in pairwise preferences. Lower Triangle: agreement in rankings. (c) t-SNE embedding of measure τ similarity. Measures close together induce similar system rankings.

significant difference [35, 36, 2]. For each model pair and metric, we conduct a significance test using the bootstrap method; we use 10,000 replicates and obtain a two-sided p-value via the centered-statistic bootstrap. We use $\alpha = 0.05$ and correct for multiple comparisons within each metric using both family-wise error rate (Holm's method) and false discovery rate (Benjamini-Hochberg). The family-wise error rate (FWER) controls the probability of making any false positive across all pairwise tests, yielding a conservative estimate of discriminative power. In contrast, the false discovery rate (FDR) controls the expected proportion of false positives among the rejected hypotheses, providing a less conservative but higher-sensitivity view.

Data efficiency ensures the measure achieves reliable and discriminative conclusions using as few evaluation samples as possible. We analyze the data efficiency of a model by measuring how quickly it converges to a stable ranking of systems as a function of evaluation examples. Specifically, we randomly downsample evaluation examples, compute the aggregated preference between system pairs, and then compute the accuracy of those preferences with respect to preferences based on the full data. We downsample 100 times at ten points. In addition to measuring how quickly system preferences converge, we measure how well a measure recovers the oracle system preferences as a function of number of examples, providing us with a sense of how quickly a measure converges to the correct preferences.

5 Results

Given the breadth of experiments, we include cross-benchmark aggregated results or benchmark-specific results. Disaggregated and complete results can be found in Appendix C (Validity), D (Reliability), E (Sensitivity), and F (Data Efficiency).

5.1 Validity

Our results provide evidence consistent with convergent validity (high correlation with existing performance metrics) and criterion validity (improving alignment with oracle preferences) of trajectory-aware preferences.

Inter-measure similarity. Figure 3a provides an example bump chart showing how a system ranking changes across different evaluation measures. As expected, SR collapses several systems into tied ranks, which non-binary measures (PR, SPL, LR, RPP, IPP) disentangle. While these measures generally agree on the resulting ordering, they tend to disagree on individual rank swaps. Figure 3b scales this inter-measure similarity analysis over all of our datasets, using pairwise preference agreement (upper triangle) and Kendall's τ of system rankings (lower triangle) to compare all six measures, resulting in two clusters. A scalar success-like family (SR, PR, SPL) that compares trajectories by an aggregate scalar outcome, and a trajectory-preference family (LR, RPP, IPP) that

	(a) Oracle Rank Acc.			(b) Split Half		(c) LOO	(d) Tie Rate	(e) Disc. Power		(f) Disc. Bias	
	Acc	FW	FD	P	R			FW	FD	FW	FD
SR	12.8	0	0	0.75	0.73	**0**	74.9	44.98	58.47	**0**	**0**
PR	12.8	0	0	0.82	0.82	2.14	49.71	56.53	73.37	**0**	**0**
SPL	91.8	31.6	48.1	0.71	0.7	5.5	63.42	39.02	60.2	**0**	**0**
LR	95.6	37.5	61.6	**0.83**	**0.85**	**0**	**33.9**	**66.16**	77.81	**0**	**0**
RPP	94.2	37.5	**63.2**	**0.83**	**0.85**	1.88	34.82	61.51	**78.35**	**0**	**0**
IPP	**95.8**	**37.7**	62.5	0.78	0.81	2.83	35.09	53.34	73.76	**0**	**0**

Table 2: Summary of meta-evaluation results across benchmarks. **Validity:** (a) Recovering oracle preferences. **Reliability:** (b) split-half reliability of system pairs (P) and system rankings (R). (c) Leave-one-out sign flip rate. **Sensitivity:** (d) Instance level tie rate. (e) Discriminative power with correction for family-wise error rate (FW) and false discovery rate (FD). (f) Discriminative bias detection of significant differences amongst identical models. Lower is better for LOO, tie rate, and discriminative bias; higher is better otherwise. Bold = best per column. Details on meta-evaluation methods Section 4.3.

compares them across return levels. We can visualize this clustering by using the τ correlation to construct a t-SNE embedding of measures (Figure 3c).

Oracle agreement. While the convergent validity captured by our inter-measure similarity analysis provides evidence of consistency between preference-based evaluation and existing metric-based evaluation methods, agreement with oracle preferences between systems (SGRL-oracle) provides criterion validity. Table 2a reports the fraction of pairwise preferences among the degraded oracle variants that each measure recovers correctly. SR and PR both achieve only 12.8% accuracy since their terminal-only character means they cannot distinguish successful agents whose differences appear only in how they reach the goal. SPL recovers 91.8% of the true preferences, while the trajectory-preference measures all exceed 94%. Requiring that oracle agreement be statistically significant demonstrates the advantage of trajectory-level preferences. Under FDR correction, RPP detects 63.2% of true preferences as significant, followed by IPP (62.5%) and LR (61.6%); SPL is only able to detect 48.1%, while SR and PR detect none. The same ordering of preferences over metrics holds under the more conservative FWER correction.

5.2 Reliability

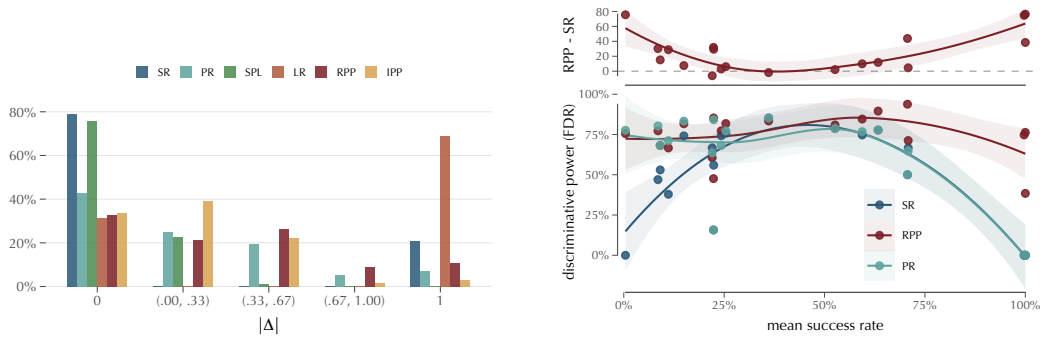
Our results demonstrate that trajectory-aware preferences produce more stable results under resampling, with fewer reversals when data is perturbed or subsampled when compared with existing metric-based evaluations.

Split-half reliability. Table 2b shows our results for split-half reliability. LR and RPP both achieve high mean split-half correlations (system pair: 0.83; system ranking: 0.85), with PR also achieving correlations above 0.80. While IPP reaches a lower correlation compared to other trajectory measures, the remaining scalar measures show notably lower reliability with correlations between 0.70 and 0.75 across system pair and system ranking. The trajectory-preference family produces more stable rankings under subsampling than the metric family, with PR and IPP sitting between the two.

Leave-one-out stability. Table 2c shows our results for leave-one-out reliability. SR and LR both achieve a 0% sign-flip rate: removing any single instance never reverses any pairwise system preference, primarily because removing an example will, in the worst case, turn a signed difference into a tie, not a flip. Of measures with less discrete behavior, RPP maintains modest sign flips (1.88%), followed by PR (2.14%) and IPP (2.83%). SPL shows the highest flip rate (5.5%), consistent with its lower split-half correlation.

5.3 Sensitivity

Our results show that, by converting many ties into informative comparisons, trajectory-aware metrics recover signal that scalar measures discard, resulting in more detectable system differences.



(a) Measure Distribution (AB)

(b) FDR correction

Figure 4: **Sensitivity:** (a) Distribution of measure values. (b) Discriminative power as a function of mean task success rate. Each point is one benchmark task. Curves are LOWESS fits. LR and IPP track RPP while SPL tracks SR; lines removed for clarity.

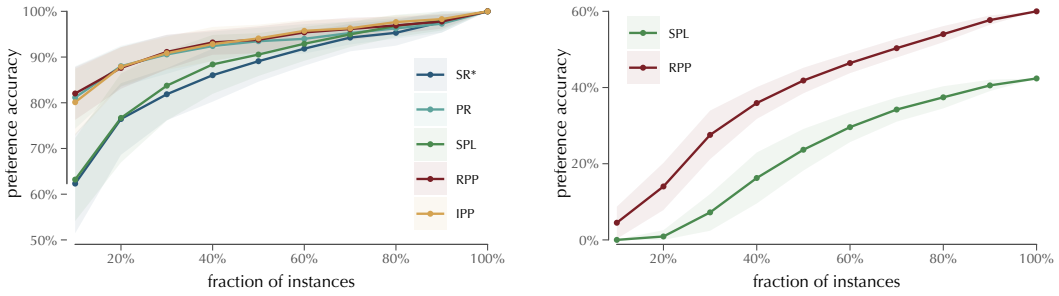
Tie rate. Table 2d shows the tie rates across measures. As mentioned in Section 1, SR produces ties for 74.9% of instance comparisons, while SPL reduces ties to 63.4% by also penalising trajectory length for successful trajectories. PR, on the other hand, mitigates this to 49.7% by using terminal progress rather than binary success. The trajectory-preference family achieves substantially lower tie rates of roughly 35%.

To understand tie rate, we can look at the distribution of Δ values for different measures. Figure 4(a) shows the distribution of the absolute value of pairwise preference values at the instance level for each measure on AgentBoard (distributions for all benchmarks can be found in Appendix Figure 7). SR and LR concentrate nearly all of their mass at the two extremes: $|\Delta| = 0$ (ties) and $|\Delta| = 1$ (maximum preference). This binary character means that individual instance comparisons carry only one bit of information. SPL shifts some mass away from the endpoints but remains heavily bimodal. In contrast, PR, RPP, and IPP distribute substantial mass across the interior of $(0, 1)$, producing a richer set of preference magnitudes. This continuous spread is the mechanism behind their lower tie rates. When each instance comparison can take a range of values rather than only $\{0, 1\}$, the aggregate preference over many instances becomes a more informative statistic, leading to tighter confidence intervals and more frequent rejection of the null hypothesis.

Discriminative power. To address concerns that lower tie rates may be due to noise, we can compute the number of statistically significant differences detected. In general, we find that the lower tie rate in preference-based evaluation reflects greater discriminative power as demonstrated in Table 2e. Under FDR correction, RPP detects significant differences in 78.4% of model pairs, with LR, IPP, and PR, also above 70%; SPL reaches 60.2% and SR 58.5%. Under the more conservative FWER correction, both LR and RPP provide discriminative power above 60%, while PR, IPP, SR, and SPL consistently detect fewer differences. These results support the higher sensitivity exhibited by preference-based evaluation, reproducing results from information retrieval research.

Figure 4b shows how discriminative power varies with mean task success rate. Each point represents a single benchmark task. SR's power peaks at intermediate success rates and collapses at both extremes. When most systems either all fail or all succeed, SR cannot distinguish them. PR is able to distinguish models when they all tend to fail but, like SR, collapses as models become more successful. RPP maintains high discriminative power across the full range of task difficulty, including on the oracle domains where SR has zero power.

Discriminative bias. We find that all measures achieve zero false-positive rates on same-model pairs (TALES-AA) under both FWER and FDR correction (Table 2f), providing evidence that the increase in discriminative power is not resulting in spurious differences.



(a) Ranking stability (AB/ALFWorld)

(b) Oracle sig. accuracy (FourRooms)

Figure 5: **Data efficiency.** (a) Accuracy of model preferences based on each subsample fraction with respect to model preferences based on the full-data. (b) Fraction of oracle pairs that are both correctly ordered and statistically significant (Benjamini-Hochberg correction) as a function of the fraction of instances used. SR and PR omitted due to poor performance; LR and IPP omitted for clarity.

5.4 Data efficiency

Our results show that, because instance-level comparisons are more informative, trajectory-aware measures reach stable rankings with fewer evaluation instances and converge faster to full-data conclusions.

Ranking stability. Figure 5a shows how the system model preferences induced by each measure converge to the full-data model preferences as evaluation instances are subsampled. On AgentBoard ALFWorld (shown), as well as on other benchmarks (Appendix Figures 8–11), preferences based on trajectories plateau with as many or fewer instances when compared with preferences based on scalar metrics. This is consistent with our reliability results (Section 5.2), where trajectory measures demonstrate stability between strategically downsampled datasets.

Oracle preference recovery. Figure 5b complements this analysis with oracle-controlled data, using a stricter criterion: a pair is counted as correctly recovered only if the measure assigns the correct sign *and* the difference is statistically significant. On FourRooms (shown), RPP's advantage is apparent at small sample sizes: RPP's significant accuracy at 50–60% of instances already exceeds SPL's full-data value. Results on DoorKey and Taxi (Appendix Figures 13 and 14) exhibit the same qualitative pattern; the advantage is most pronounced on Taxi, the hardest domain, where SPL detects essentially no significant pairs while RPP and the other trajectory-preference measures still recover a non-trivial fraction. Trajectory-level information thus improves not only oracle agreement in general (Section 5.1) but also sample efficiency in detecting true system differences with statistical confidence.

6 Discussion

Our results provide evidence that preference-based approaches can improve reliability, sensitivity, and data efficiency while preserving alignment with existing performance measures and improving agreement with oracle preferences without requiring additional data beyond trajectory logs, changes to sampling practices, reweighting, or hyper-parameters.

Our results suggest that benchmark saturation, often portrayed as the result of poor data collection or weak problem difficulty, may also be explained by the choice of evaluation measure. Figure 4b suggests that a measure like SR may be effective at distinguishing 'middling' systems but fail altogether at early points in the development process (when models may be largely under-performant) or later in the development process (when models may be uniformly strong). The intentional design of sensitive metrics allows a benchmark to more effectively compare arbitrary populations of models.

Our adoption of preference-based evaluation, while common in online or arena-style evaluation, is novel for offline evaluation outside of simple paired statistical tests. Existing studies in production evaluation demonstrate the effectiveness and efficiency of preference-based evaluation [5]. Beyond this, offline preference-based win rates are comparable with arena-style win rates, allowing more

consistent evaluation and avoiding any cross-metric calibration [24]. At the same time, offline preference-based evaluation presents the opportunity for counterfactual preference measurement, which is impossible in online evaluation where a real user is often limited to comparing two system outputs.

Working with temporal preference instead of temporal discounting allows our measures to avoid needing to validate a precise relationship between time and utility (or return). We only require that the preference be consistent across test instances without any hyperparameters.

Finally, while we have focused on preference-based evaluation, all of our measures suggest novel methods for optimizing sequential decision-making tasks. Avoiding the need to select a discount factor, craft partial rewards, or worry about consistent cross-task temporal discounting may allow the more efficient and robust training of models. There is increasing evidence that preferences can be more expressive than methods that reduce performance to a scalar metric value [29, 39].

Limitations Trajectory-aware evaluation assumes that intermediate returns reflect meaningful progress toward task completion; when subgoal annotations are noisy, weakly calibrated, artificially dense, or poorly aligned with human notions of utility, preference-based metrics may amplify annotation artifacts rather than genuine performance differences. In addition, in some domains, temporal preference, while embedded in the reinforcement learning and economics literatures, may not be a desirable system property. Finally, because the strongest oracle-ranking analyses rely on synthetic environments with known optimal behavior, further work is needed to validate the robustness of these findings in real-world agentic systems with imperfect or latent reward structure.

7 Conclusion

We argued that success rate as a metric discards information, compromises the efficiency of benchmarks, and leads to benchmark saturation. By shifting to preference-based comparisons over trajectory structure, we recover this lost signal without requiring additional data beyond trajectory logs or stronger assumptions about the relationship between time and utility. Empirically, this yields consistent gains in reliability, sensitivity, and data efficiency across benchmarks. More broadly, our results suggest that evaluation quality and benchmark utility often depend on the measurement instrument itself.

References

[1] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf`.

[2] M. Akhtar, A. Reuel, P. Soni, S. Ahuja, P. S. Ammanamanchi, R. Rawal, V. Zouhar, S. Yadav, C. Whitehouse, D. Ki, J. Mickel, L. Choshen, M. Šuppa, J. Batzner, J. Chim, J. Sania, Y. Long, H. A. Rahmani, C. Knight, Y. Nan, J. Raj, Y. Fan, S. Singh, S. Sahoo, E. Habba, U. Gohar, S. Pawar, R. Scholz, A. Subramonian, J. Ni, M. Kochenderfer, S. Koyejo, M. Sachan, S. Biderman, Z. Talat, A. Ghosh, and I. Solaiman. When ai benchmarks plateau: A systematic study of benchmark saturation. In *Forty-third International Conference on Machine Learning*, 2026.

[3] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. URL `http://arxiv.org/abs/1807.06757`.

[4] S. Ashury Tahan, A. Gera, B. Sznajder, L. Choshen, L. Ein-Dor, and E. Shnarch. Label-efficient model selection for text generation. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8384–8402, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.456. URL `https://aclanthology.org/2024.acl-long.456/`.

[5] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30(1), Mar. 2012. ISSN 1046-8188. doi: 10.1145/2094072.2094078. URL https://doi.org/10.1145/2094072.2094078.

[6] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M. I. Jordan, J. E. Gonzalez, and I. Stoica. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[7] A. Chouldechova, A. F. Cooper, S. Barocas, A. Palia, D. Vann, and H. Wallach. Comparison requires valid measurement: Rethinking attack success rate comparisons in AI red teaming. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*, 2025. URL https://openreview.net/forum?id=d7hqAhLvWG.

[8] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988.

[9] C. Z. Cui, X. Yuan, Z. Xiao, P. Ammanabrolu, and M.-A. Côté. Tales: Text adventure learning environment suite, 2025. URL https://arxiv.org/abs/2504.14128.

[10] F. Diaz and A. Ferraro. Offline retrieval evaluation without evaluation metrics. In *Proceedings of the 45th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 599–609, New York, NY, USA, 2022. Association for Computing Machinery. URL https://doi.org/10.1145/3477495.3532033.

[11] F. Diaz, M. D. Ekstrand, and B. Mitra. Recall, robustness, and lexicographic evaluation. *ACM Trans. Recomm. Syst.*, 4(1), July 2025. doi: 10.1145/3728373. URL https://doi.org/10.1145/3728373.

[12] S. Frederick, G. Loewenstein, and T. O'Donoghue. Time discounting and time preference: A critical review. *Journal of Economic Literature*, 40(2):351–401, June 2002. doi: 10.1257/002205102320161311. URL https://www.aeaweb.org/articles?id=10.1257/002205102320161311.

[13] A. Ghosh, Y. Mai, G. Channing, and L. Choshen. AI evals are becoming the new compute bottleneck. EvalEval Coalition Blog, Apr. 2026. URL https://evalevalai.com/research/2026/04/29/eval-costs-bottleneck/.

[14] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11694. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

[15] Y. Huang, J. Song, Q. Hu, F. Juefei-Xu, and L. Ma. Actracer: Active testing of large language model via multi-stage sampling. *ACM Trans. Softw. Eng. Methodol.*, 35(3), Feb. 2026. ISSN 1049-331X. doi: 10.1145/3744340. URL https://doi.org/10.1145/3744340.

[16] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002. ISBN 1-58113-567-X. doi: http://doi.acm.org/10.1145/775047.775067.

[17] S. Kapoor, B. Stroebl, P. Kirgis, N. Nadgir, Z. S. Siegel, B. Wei, T. Xue, Z. Chen, F. Chen, S. Utpala, F. Ndzomga, D. Oruganty, S. Luskin, K. Liu, B. Yu, A. Arora, D. Hahm, H. Trivedi, H. Sun, J. Lee, T. Jin, Y. Mai, Y. Zhou, Y. Zhu, R. Bommasani, D. Kang, D. Song, P. Henderson, Y. Su, P. Liang, and A. Narayanan. Holistic agent leaderboard: The missing infrastructure for AI agent evaluation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vUaY1t64ZZ.

[18] D. Kiela, M. Bartolo, Y. Nie, D. Kaushik, A. Geiger, Z. Wu, B. Vidgen, G. Prasad, A. Singh, P. Ringshia, Z. Ma, T. Thrush, S. Riedel, Z. Waseem, P. Stenetorp, R. Jia, M. Bansal, C. Potts, and A. Williams. Dynabench: Rethinking benchmarking in NLP. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, pages 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

[19] J. Kossen, S. Farquhar, Y. Gal, and T. Rainforth. Active testing: Sample-efficient model evaluation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5753–5763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kossen21a.html.

[20] Y. Li, J. Ma, M. Ballesteros, Y. Benajiba, and G. Horwood. Active evaluation acquisition for efficient LLM benchmarking. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EHqQaBYYlE.

[21] C. Ma, J. Zhang, Z. Zhu, C. Yang, Y. Yang, Y. Jin, Z. Lan, L. Kong, and J. He. Agentboard: An analytical evaluation board of multi-turn llm agents. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 74325–74362. Curran Associates, Inc., 2024. doi: 10.52202/079017-2365. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/877b40688e330a0e2a3fc24084208dfa-Paper-Datasets_and_Benchmarks_Track.pdf.

[22] Z. Ma, K. Ethayarajh, T. Thrush, S. Jain, L. Wu, R. Jia, C. Potts, A. Williams, and D. Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10351–10367. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/55b1927fdafef39c48e5b73b5d61ea60-Paper.pdf.

[23] F. Maia Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, and M. Yurochkin. tinybenchmarks: evaluating llms with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[24] A. Maksai, F. Garcin, and B. Faltings. Predicting online performance of news recommender systems through richer evaluation metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 179–186, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336925. doi: 10.1145/2792838.2800184. URL https://doi.org/10.1145/2792838.2800184.

[25] J. Mandel and R. D. Stiehler. Sensitivity–a criterion for the comparison of methods of test. *Journal of research of the National Bureau of Standards*, 53:155, 1954. URL https://api.semanticscholar.org/CorpusID:52393909.

[26] F. Martínez-Plumed, R. B. Prudêncio, A. Martínez-Usó, and J. Hernández-Orallo. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18–42, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.09.004. URL https://www.sciencedirect.com/science/article/pii/S0004370219300220.

[27] A. Moffat and J. Mackenzie. How much freedom does an effectiveness metric really have? *Journal of the Association for Information Science and Technology*, n/a(n/a), 2024. doi: https://doi.org/10.1002/asi.24874. URL https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24874.

[28] A. K. Mohankumar and M. Khapra. Active evaluation: Efficient NLG evaluation with few pairwise comparisons. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8761–8781, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.600. URL https://aclanthology.org/2022.acl-long.600.

[29] R. Munos, M. Valko, D. Calandriello, M. G. Azar, M. Rowland, Z. D. Guo, Y. Tang, M. Geist, T. Mesnard, C. Fiegel, A. Michi, M. Selvi, S. Girgin, N. Momchev, O. Bachem, D. J. Mankowitz, D. Precup, and B. Piot. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Y5AmNYiyCQ.

[30] F. Ndzomga. Efficient benchmarking of ai agents, 2026. URL https://arxiv.org/abs/2603.23749.

[31] A. Olteanu, S. L. Blodgett, A. Balayn, A. Wang, F. Diaz, F. du Pin Calmon, M. Mitchell, M. Ekstrand, R. Binns, and S. Barocas. Rigor in ai: Doing rigorous ai work requires a broader, responsible ai-informed conception of rigor. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.14652.

[32] S. Ott, A. Barbosa-Silva, K. Blagec, J. Brauner, and M. Samwald. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13 (1):6793, 2022. doi: 10.1038/s41467-022-34591-0. URL https://doi.org/10.1038/s41467-022-34591-0.

[33] M. Peyrard, W. Zhao, S. Eger, and R. West. Better than average: Paired evaluation of NLP systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2301–2315, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.179. URL https://aclanthology.org/2021.acl-long.179.

[34] P. Rodriguez, J. Barrow, A. Hoyle, J. P. Lalor, R. Jia, and J. Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.346. URL https://aclanthology.org/2021.acl-long.346/.

[35] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 525–532, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933697. doi: 10.1145/1148170.1148261. URL https://doi.org/10.1145/1148170.1148261.

[36] T. Sakai. Alternatives to bpref. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 71–78, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: http://doi.acm.org/10.1145/1277741.1277756.

[37] N. Subramani, A. Gomez, and M. T. Diab. SimBA: Simplifying benchmark analysis using performance matrices alone. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13220–13233, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.711. URL https://aclanthology.org/2025.findings-emnlp.711/.

[38] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, 1998.

[39] G. Swamy, C. Dann, R. Kidambi, S. Wu, and A. Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 47345–47377. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/swamy24a.html.

[40] O. Team. Openhands index: A comprehensive leaderboard for ai coding agents. https://index.openhands.dev, 2025.

[41] S. T. Truong, Y. Tu, P. Liang, B. Li, and S. Koyejo. Reliable and efficient amortized model-based evaluation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=HDbWrsgkB9.

Year	ACL	EMNLP	NeurIPS Main	NeurIPS Data
2022	8.8%	8.9%	13.6%	4.9%
2023	10.5%	10.9%	13.5%	7.8%
2024	10.4%	12.6%	13.0%	11.1%
2025	17.0%	20.5%	18.4%	18.3%

Table 3: Percentage of published papers abstracts that reference binary metrics.

[42] C. Vania, P. M. Htut, W. Huang, D. Mungra, R. Y. Pang, J. Phang, H. Liu, K. Cho, and S. R. Bowman. Comparing test sets with item response theory. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.92. URL https://aclanthology.org/2021.acl-long.92/.

[43] M. N. Volkovs and R. S. Zemel. A flexible generative model for preference aggregation. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 479–488, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312295. doi: 10.1145/2187836.2187902. URL https://doi.org/10.1145/2187836.2187902.

[44] H. Wallach, M. Desai, A. F. Cooper, A. Wang, C. Atalla, S. Barocas, S. L. Blodgett, A. Chouldechova, E. Corvi, P. A. Dow, J. Garcia-Gathright, A. Olteanu, N. J. Pangakis, S. Reed, E. Sheng, D. Vann, J. W. Vaughan, M. Vogel, H. Washington, and A. Z. Jacobs. Position: Evaluating generative AI systems is a social science measurement challenge. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL https://openreview.net/forum?id=1ZC4RNjqzU.

[45] F. F. Xu, Y. Song, B. Li, Y. Tang, K. Jain, M. Bao, Z. Z. Wang, X. Zhou, Z. Guo, M. Cao, M. Yang, H. Y. Lu, A. Martin, Z. Su, L. Maben, R. Mehta, W. Chi, L. Jang, Y. Xie, S. Zhou, and G. Neubig. Theagentcompany: Benchmarking llm agents on consequential real world tasks, 2024. URL https://arxiv.org/abs/2412.14161.

A Use of binary metrics at ML and NLP conferences

We used the OpenReview API to gather abstracts for NeurIPS, NeurIPS Datasets and Benchmarks, ACL, and EMNLP between 2022 and 2025. We then identified abstracts that contained references to any of: success rate, accuracy, exact match, task success, episode success, top-1 accuracy, pass@1, solved rate, or match rate. Complete results are presented in Table 3.

B Datasets

Datasets consist of agent trajectories on benchmark tasks. AgentBoard data [21] downloaded from https://huggingface.co/datasets/hkust-nlp/agentboard/resolve/main/data.tar.gz. OpenHands Index data [40] downloaded on 11 April 2026 for all runs in https://github.com/OpenHands/openhands-index-results. TheAgentCompany data [45] downloaded from https://github.com/TheAgentCompany/experiments/tree/main/evaluation/1.0.0. Text Adventure Learning Environment Suite data [9] downloaded on 11 April 2026 from https://huggingface.co/datasets/PEARLS-Lab/TALES-Trajectories. To support statistical analysis, we remove tasks with fewer than 30 task instances.

B.1 Sub-Goal Reinforcement Learning

The sub-goal reinforcement learning data instantiates three classic gridworld-style domains: Taxi (the Gymnasium Taxi-v3 environment), DoorKey (the MiniGrid DoorKey-5x5-v0 environment, in which the agent must pick up a key, unlock a door, and reach a goal cell), and FourRooms (the MiniGrid FourRooms environment, in which four rooms are connected by single-cell hallways and the agent must reach a goal placed in another room). For each domain we construct a fixed bank of

Task	Models	Instances	Avg. length	Avg. density	Partial	Interm.	License
AgentBoard (AB)							GPL-2.0
alfworld	12	134	2.78	0.1920	✓	✓	
babyai	12	112	2.73	0.1336	✓	✓	
pddl	12	60	5.55	0.1146	✓	✓	
scienceworld	12	90	3.70	0.1340	✓	✓	
tool-query	12	60	3.31	0.9720	✓	✓	
webshop	12	251	4.58	0.7432	✓	✓	
Openhands-Index (OHI)							MIT
gaia	22	165	19.87	0.0821	—	—	
swe-bench	21	500	67.04	0.0169	—	—	
swe-bench-mm	21	103	87.94	0.0037	—	—	
swt-bench	18	433	47.74	0.0183	—	—	
TheAgentCompany (TAC)							N/A
TAC	16	175	26.27	0.0380	✓	—	
Text Adventure Learning Environment Suite (TALES)							N/A
jericho	54	55	36.45	0.0980	✓	✓	
scienceworld	53	30	32.25	0.2979	✓	✓	
Sub-Goal Reinforcement Learning (SGRL)							N/A
doorkey	30	48	8.23	0.2074	✓	✓	
fourrooms	26	100	9.13	0.7198	✓	✓	
taxi	18	100	10.71	0.1286	✓	✓	

Table 4: Dataset statistics. Avg. length: mean of max step index per trajectory. Avg. density: mean fraction of steps with a new return increase. Partial: any non-binary rewards present. Interm.: any trajectory has a reward between the start and final returns at an intermediate step.

up to 100 distinct task instances by iterating reset seeds from zero upward and retaining only the first seed whose post-reset state, characterized by a domain-specific tuple of task-defining factors, is novel relative to all previously retained instances; the discriminating tuple is (taxi row, taxi column, passenger location, destination) for Taxi, (agent position, agent heading, key position, door position, door-locked flag, goal position) for DoorKey, and (agent position, agent heading, goal position, goal room) for FourRooms. Each retained seed is serialized together with its decoded state, and at evaluation time the environment is reset with the stored seed and the recorded factors are asserted to match, which keeps the bank reproducible across runs. The DoorKey domain has 48 instances because the space was exhausted.

From this fixed bank we generate trajectories for a ladder of policies that span weak to strong on each domain. Two reference policies are hand-coded: a uniform random agent over the legal action set, and a deterministic oracle implemented as a planner with full environment knowledge (a shortest-path policy over Taxi's known transition graph, and a breadth-first search over (x, y, heading) tuples that emits the relative turn/forward/pickup/toggle action sequences needed for the two MiniGrid domains). The remaining systems are learned with standard model-free RL trained from sparse environment reward, with one seed-per-checkpoint and three (Taxi) or two (DoorKey, FourRooms) random seeds: for Taxi we use action-masked PPO, DQN, and QRDQN over a factored one-hot symbolic observation (decomposed into taxi row, taxi column, passenger location, and destination index), trained for 200k, 200k, and 50k steps respectively; for DoorKey and FourRooms we use PPO and A2C with a CNN policy over the default partial-observability image observation, trained for 500k and 2M steps respectively, together with a deliberately under-budgeted "weak" PPO (100k for DoorKey, 500k for FourRooms, with a 32-dim feature extractor) to populate the lower end of the performance ladder. To probe the effect of denser learning signal on the same algorithms, each learned baseline is duplicated as a "shaped" variant that trains the same architecture and hyperparameters under a potential-based shaping reward computed from the symbolic state: the Taxi shaping rewards moving toward the passenger and then toward the destination, while the MiniGrid shapings reward progress toward the next subgoal in the canonical subgoal chain (key, door, goal for DoorKey; goal-room entry then goal cell for FourRooms). At evaluation time every policy is rolled out once per banked instance under a per-domain step budget (100 for Taxi, 150 for the MiniGrid domains), and we save compact per-step logs (the decoded factored state for Taxi; agent position, carried-object

	Accuracy				Power (FDR)			
	Taxi	DoorKey	FourRooms	Mean	Taxi	DoorKey	FourRooms	Mean
SR	0	0	38.4	12.8	0	0	0	0
PR	0	0	38.4	12.8	0	0	0	0
SPL	82.1	97.4	95.8	91.8	1.6	77.9	64.7	48.1
LR	92.1	98.4	96.3	95.6	34.7	75.3	74.7	61.6
RPP	88.9	97.4	96.3	94.2	38.4	76.3	74.7	63.2
IPP	92.6	98.4	96.3	95.8	34.7	77.9	74.7	62.5

Table 5: Oracle rank accuracy (%) and power (%) (C(20,2)=190 pairs per domain).

type, and door-open flag for DoorKey; agent position and current room identifier for FourRooms) so that hidden subgoal-progress labels can be recovered offline by a deterministic detector—pickup and successful drop-off for Taxi, key pickup, door opening, and goal arrival for DoorKey, and entering the goal room and reaching the goal cell for FourRooms—producing a return-jump trace that assigns equal credit to each subgoal of a domain.

The oracle-degradation suite holds the instance bank, the oracle planner, and the random seeding fixed, and rolls out a wrapper policy that, at each step independently, replaces the oracle's chosen action with an action drawn uniformly at random from the full action space with probability ϵ; we then sweep ϵ using a two-phase calibration in which a coarse grid first brackets the value at which mean episode-return drops to roughly 80% of oracle return, after which 19 ϵ values are chosen linearly spaced from a small lower bound to that 80% crossing (approximately 0.0003 to 0.0057 for Taxi, 0.010 to 0.190 for FourRooms, and 0.025 to 0.475 for DoorKey), producing for each domain a set of twenty closely spaced systems—the oracle plus nineteen degraded variants—whose performance differences are small enough to stress the metrics under study.

C Validity

C.1 Bump Charts

Figure 6 traces each model's time-to-return rank across measures, one panel per task. Crossings reveal where the rankings are reordered; flat bands indicate stable rankings.

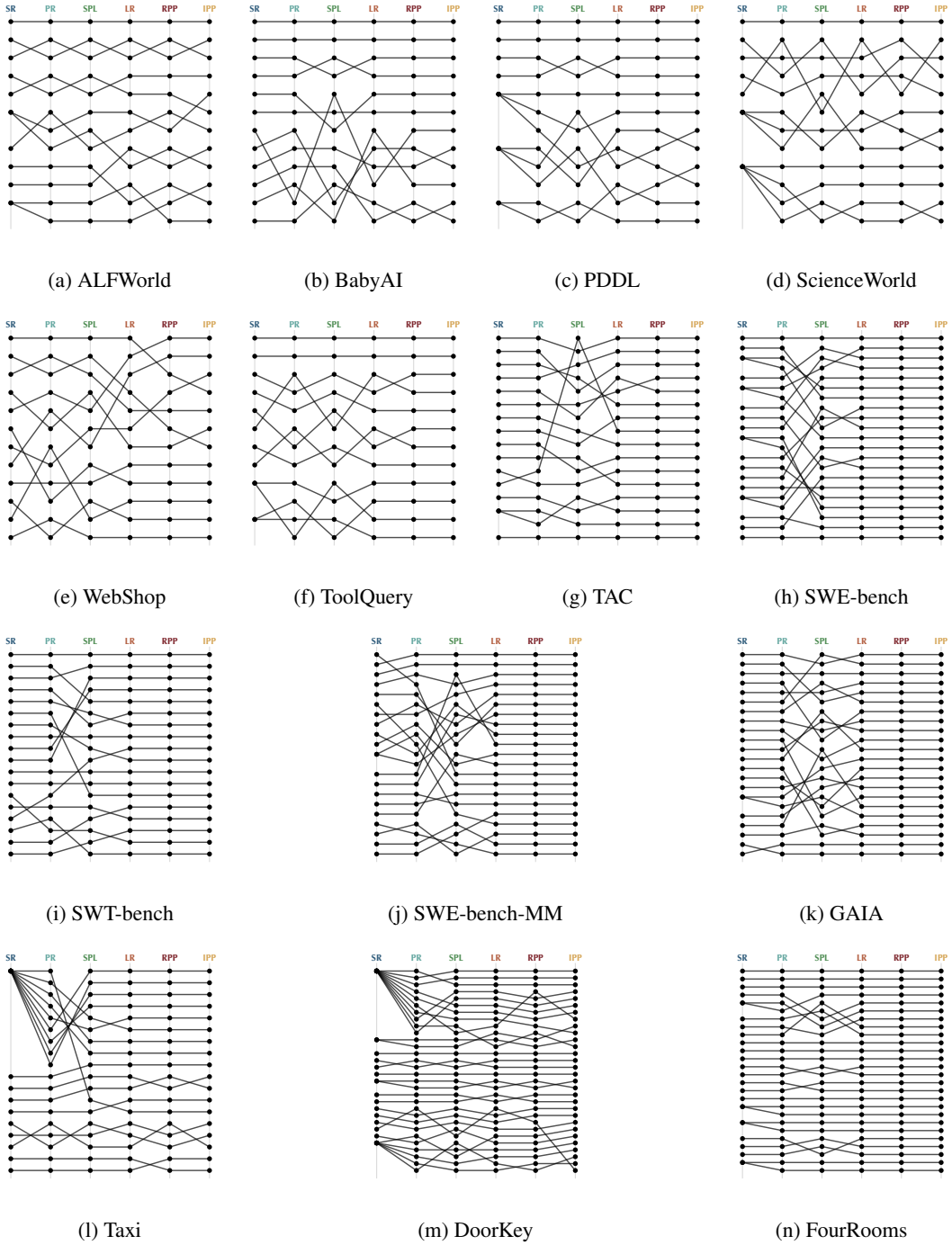
C.2 Oracle Rank Accuracy

To measure whether each metric predicts the correct ordering between systems with known ground-truth performance, we construct 20 variants of an oracle agent per domain by injecting ε-random actions at varying rates ($\varepsilon = 0$ for the oracle, increasing to $\approx 80\%$ of oracle episode return at $\varepsilon_{\max}$). This yields $\binom{20}{2} = 190$ ordered pairs per domain (570 total), where the correct ordering is defined by ε: a lower-noise agent is always better.

Table 5 reports rank accuracy (fraction of pairs ranked correctly) and statistical power (fraction of correctly ranked pairs that are also significant at $\alpha = 0.05$ by two-sided bootstrap test).

D Reliability

Table 6 reports split-half reliability per metric and benchmark: instances are split into halves and we compare both per-instance scores and the induced system rankings across the two halves. Table 7 complements this with a leave-one-out stress test, giving the fraction of system pairs whose sign of difference flips when any single instance is dropped—a direct measure of how brittle pairwise comparisons are at each benchmark's current size.



(a) ALFWorld (b) BabyAI (c) PDDL (d) ScienceWorld

(e) WebShop (f) ToolQuery (g) TAC (h) SWE-bench

(i) SWT-bench (j) SWE-bench-MM (k) GAIA

(l) Taxi (m) DoorKey (n) FourRooms

Figure 6: Bump charts for tasks. Each line is one model; rank 1 is top.

	System pairs						System ranking					
	AB	TAC	OHI	TALES	SGRL	Mean	AB	TAC	OHI	TALES	SGRL	Mean
SR	0.82	0.81	0.66	0.55	0.92	0.75	0.79	0.84	0.68	0.4	0.93	0.73
PR	0.82	0.87	0.66	0.84	0.9	0.82	0.76	0.89	0.68	0.85	0.92	0.82
SPL	0.74	0.63	0.79	0.52	0.88	0.71	0.74	0.68	0.79	0.39	0.9	0.7
LR	0.82	0.8	0.79	0.84	0.91	0.83	0.78	0.85	0.82	0.89	0.91	0.85
RPP	0.83	0.79	0.79	0.82	0.91	0.83	0.81	0.87	0.82	0.85	0.92	0.85
IPP	0.81	0.79	0.79	0.62	0.9	0.78	0.8	0.87	0.82	0.62	0.91	0.81

Table 6: Split-half reliability (instance-level and system-ranking), averaged within benchmark.

	AB	TAC	OHI	TALES	SGRL	Mean
SR	0	0	0	0	0	0
PR	4.55	0.83	0	4.89	0.45	2.14
SPL	6.57	10	5.89	3.03	2.01	5.5
LR	0	0	0	0	0	0
RPP	2.53	0.83	0	5.21	0.82	1.88
IPP	3.54	0.83	0	9.32	0.46	2.83

Table 7: Leave-one-out sign flip rate (% of system pairs where dropping one instance changes sign of difference), averaged within benchmark.

	AB	TAC	OHI	TALES	SGRL	Mean
SR	78.88	80.88	74.56	84.05	56.1	74.9
PR	43.49	49.78	74.56	27.61	53.08	49.71
SPL	75.87	76.01	36.77	80.73	47.74	63.42
LR	33.27	32.3	36.77	22.66	44.49	33.9
RPP	34.93	34.75	36.77	23.04	44.63	34.82
IPP	36.11	34.75	36.77	23.18	44.66	35.09

Table 8: Tie rate (% of instance comparisons with zero difference), averaged within benchmark.

E Sensitivity

E.1 Tie rate

Table 8 reports the fraction of instance-level comparisons in which two system outputs receive identical scores, averaged within each benchmark. High tie rates indicate a metric with limited resolution—many pairs of systems are indistinguishable on a given instance, which weakens its ability to support fine-grained ranking.

E.2 Preference Distributions

Figure 7 shows the distribution of absolute pairwise preference values $|\Delta|$ for each evaluation metric, broken down by benchmark group. Each bar shows the proportion of instance-level comparisons whose absolute preference falls in the indicated range. The point masses at 0 (ties) and 1 (maximum disagreement) are shown as separate bars; the three interior bins partition $(0, 1)$ into equal thirds.

E.3 Discriminative Power

Table 9 reports the fraction of system pairs whose score difference is statistically significant under a bootstrap test, averaged within each benchmark. Higher values mean the metric resolves more pairs of systems—a complementary view to the tie rate above, now accounting for sampling variability rather than just exact ties.

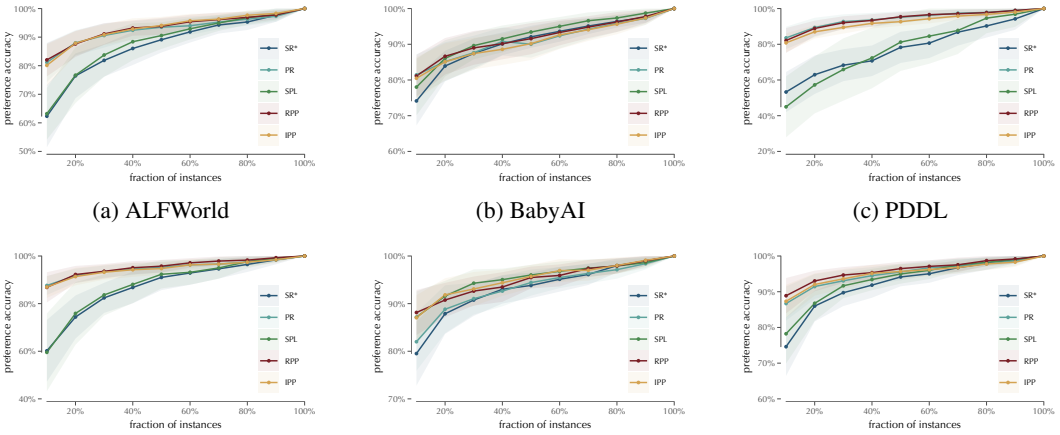
	FWER						**FDR**					
	AB	TAC	OHI	TALES	SGRL	Mean	AB	TAC	OHI	TALES	SGRL	Mean
SR	47.22	58.33	38.6	15.97	64.77	44.98	59.09	74.17	55.57	27.98	75.54	58.47
PR	62.37	73.33	38.6	42.58	65.77	56.53	71.46	83.33	55.57	80.91	75.57	73.37
SPL	40.15	27.5	59.71	4.5	63.21	39.02	55.81	64.17	77.08	25.87	78.09	60.2
LR	61.11	69.17	63.74	67.98	68.8	66.16	69.44	79.17	77.98	83.8	78.68	77.81
RPP	63.64	66.67	63.74	44.18	69.32	61.51	71.97	81.67	77.98	80.41	79.74	78.35
IPP	61.62	66.67	63.74	5.52	69.17	53.34	71.21	81.67	77.98	58.74	79.2	73.76

Table 9: Discriminative power (% of pairs with significant difference, bootstrap test), averaged within benchmark.

(a) AgentBoard (b) OpenHands (c) TheAgentCompany

(d) TALES (e) TALES-AA (f) SGRL

Figure 7: Preference distributions across all benchmark datasets.



(a) ALFWorld (b) BabyAI (c) PDDL

(d) ScienceWorld (e) WebShop (f) ToolQuery

Figure 8: Preference preservation (self-reference) — AgentBoard tasks.

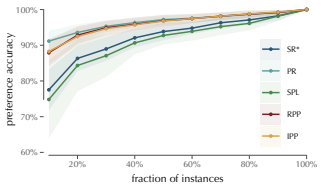
F Data efficiency

F.1 Stability

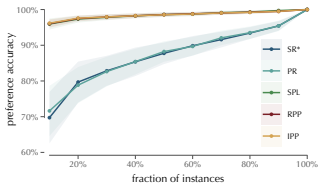
Figures 8–11 show how each metric's pairwise preferences degrade as the evaluation budget shrinks: at each subsample fraction, we measure how often the preference computed on the subset agrees with the preference computed on the full instance set (self-reference). Curves that stay near 1 indicate a measure whose system preferences are stable under aggressive subsampling; curves that fall off quickly mean the metric needs the full benchmark to be trustworthy.

F.2 Oracle recovery

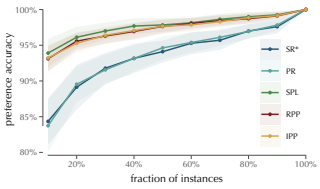
Figures 12–14 trace how reliably each metric recovers the ground-truth ordering between ε-degraded oracle variants as the fraction of instances used shrinks. Figure 12 reports raw sign accuracy (fraction of oracle pairs ordered correctly), while Figures 13 and 14 report the stricter joint criterion of correct ordering *and* statistical significance under FWER (Holm) and FDR (Benjamini-Hochberg) correction. Higher curves at small sample fractions indicate a metric that extracts a correct, defensible verdict from less data.



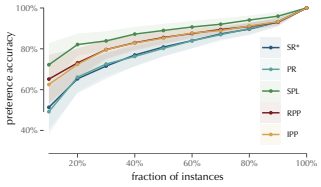
(a) TAC



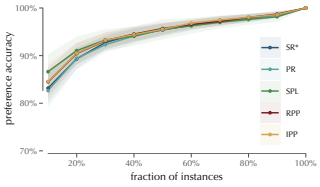
(b) SWE-bench



(c) SWT-bench

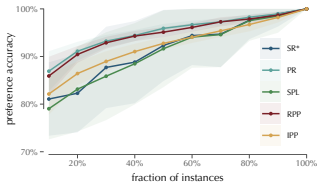


(d) SWE-bench-MM

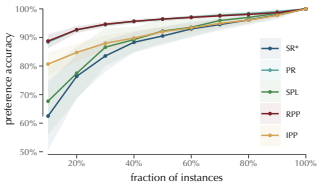


(e) GAIA

Figure 9: Preference preservation (self-reference) — TAC and OHI tasks.

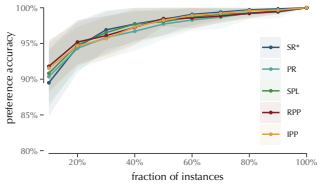


(a) Jericho

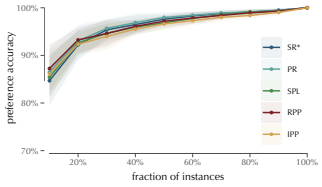


(b) ScienceWorld

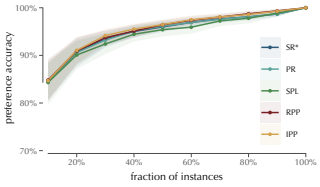
Figure 10: Preference preservation (self-reference) — TALES tasks ($\geq$ 30 instances).



(a) Taxi

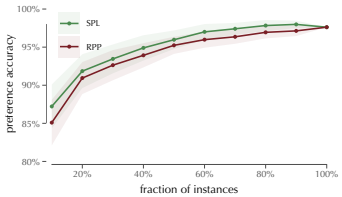


(b) DoorKey

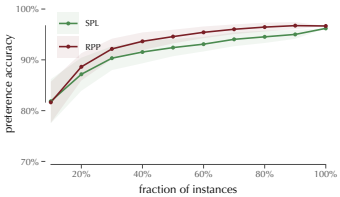


(c) FourRooms

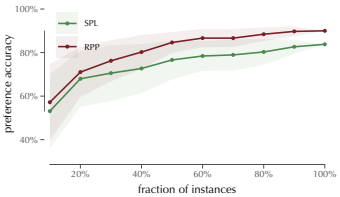
Figure 11: Preference preservation (self-reference) — SGRL tasks.



(a) DoorKey

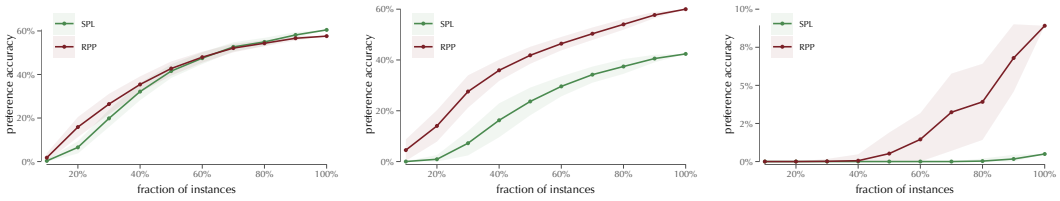


(b) FourRooms



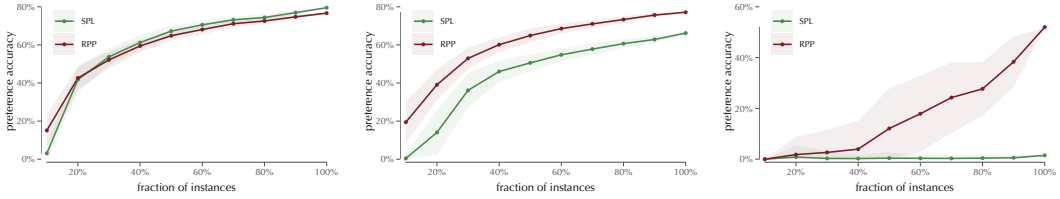
(c) Taxi

Figure 12: Oracle sign accuracy as a function of the fraction of instances used to compute each metric. Pairs are drawn from the degraded oracle variants. RPP recovers the correct pairwise preference more reliably than SPL at all sample sizes, with the advantage most pronounced on the harder Taxi domain.



(a) DoorKey (b) FourRooms (c) Taxi

Figure 13: Oracle significant accuracy (FWER, Holm correction) as a function of the fraction of instances. The fraction of oracle pairs that are both correctly ordered *and* statistically significant after family-wise error rate correction.



(a) DoorKey (b) FourRooms (c) Taxi

Figure 14: Oracle significant accuracy (FDR, Benjamini-Hochberg correction) as a function of the fraction of instances. The fraction of oracle pairs that are both correctly ordered *and* statistically significant after false discovery rate correction.