Proper Scoring Rules for Agentic Uncertainty Quantification

Suresh Raghu[*]
Independent Researcher
sureshraghu0706@gmail.com

Satwik Pandey[*]
Independent Researcher
psatwik2711@gmail.com

Shashwat Pandey
Independent Researcher
shashwat1225@gmail.com

Abstract

Language-model agents increasingly emit uncertainty signals throughout a trajectory, but existing agentic UQ evaluations often conflate ranking usefulness with probabilistic truthfulness. AUROC, AUPRC, risk–coverage, Trajectory ECE, and scalarized trajectory scores evaluate discrimination, binwise calibration, or collapsed summaries, but do not strictly elicit the full prefix-conditioned success-probability trace $q_t = \mathbb{P}^\pi(Y{=}1 \mid \mathcal{H}_t)$. Building on prequential proper scoring, we introduce the Trajectory Proper Score (TPS), a predictor-agnostic family of strictly proper trajectory-level scoring rules for any per-step uncertainty signal calibrated into a probability of eventual success. We prove that TPS strictly elicits the success-probability process under complete observation, within the chosen score family and weight schedule. We extend the construction to administratively censored trajectories by projecting the complete-data score onto the observable stopped prefix, yielding an exact q_Z-weighted reduced score and a tractable approximation when q_Z is unestimated. We further show that common trajectory evaluators target weaker objects than the full prefix-conditioned probability process: Trajectory ECE is resolution-blind, while scalarized Trajectory Brier elicits only the collapsed scalar, not the full trace. Experiments on StrategyQA, Tau2-Bench, HotpotQA, and WebShop show that these theoretical distinctions are operationally visible: probability recalibration can substantially change TPS while leaving rank metrics nearly unchanged, and the tractable censored approximation can change the verdict relative to complete-only evaluation. Code and data are available at `https://github.com/R-Suresh07/proper-scoring-agentic-uq`.

1 Introduction

Language-model agents increasingly act through extended trajectories: they observe, reason, call tools, revise plans, and eventually either succeed or fail. In such settings, uncertainty is not a property of the final answer alone. A useful agent should know, at each prefix of the interaction, how likely the task is to succeed from that point onward.

Recent agentic UQ work has moved from final-answer uncertainty toward trajectory-level signals, including stepwise calibration, propagated uncertainty, and trajectory confidence aggregation [1–4]. These methods mark an important shift, but they leave open a more basic evaluator-side question: what should count as a correct evaluation of such a trace?

We argue that the missing object is a strictly proper evaluator for the prefix-conditioned probability process. Existing trajectory evaluations typically rely on AUROC, trajectory-ECE, or scalarized trajectory-Brier. These are useful diagnostics for particular summaries: AUROC measures rank discrimination, ECE measures binwise marginal calibration, and scalar Brier scores a collapsed

[*]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

binary probability. None of them, however, strictly elicits the full process $(F_t)_{t=1}^T$, where F_t is the agent's reported probability of eventual task success from the current prefix. This distinction matters: two agents can induce similar rankings while assigning very different probabilities, and downstream interventions such as deferral, reflection, or human handoff depend on the probability scale, not only on rank.

The forecast object in this paper is deliberately simple. At step t, after observing history $\mathcal{H}_t$, the agent reports $F_t \in [0, 1]$, interpreted as the probability that the task will eventually succeed. The truthful target is

$$q_t := \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_t),$$

the continuation-success probability under the evaluation law. This process need not be monotone: a calibrated agent should lower F_t after a harmful action and raise it after acquiring useful evidence [5].

We make the following contributions.

1. We formalize agentic uncertainty evaluation as strictly proper elicitation of the prefix-conditioned success process $q_t = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_t)$, and define the weighted trajectory score
$$\mathrm{TPS}(F_{1:T}, Y) = \sum_{t=1}^{T} w_t \, S_{\alpha,\beta}(F_t, Y).$$
2. We extend the trajectory score to censored traces via conditional projection onto the observable prefix. In the binary agentic setting this reduces to an exact censored score parameterized by a single continuation-success weight q_Z.
3. We show that common trajectory evaluators target weaker objects: trajectory-ECE is resolution-blind, and scalarized trajectory-Brier is proper only for a collapsed scalar rather than the full prefix-conditioned trace.
4. Empirically, we show that recalibration can leave rank-based metrics nearly unchanged while strictly proper trajectory scores reveal large probability-scale errors, and that censored-aware scoring yields different conclusions from complete-only evaluation on administratively censored agent trajectories.

The aim of this work is to separate the question of how an agent produces uncertainty from the question of whether the reported uncertainty means what it says. TPS is agnostic to the source of the signal: verbal confidence, token-probability features, post-hoc calibrators, and predictor-side agentic UQ methods such as SAUP, UProp, and AUQ can all be evaluated once expressed as prefix-conditioned success probabilities. Section 7 summarizes how to apply TPS in practice.

2 Related Work

Strictly proper scoring rules provide the classical language for evaluating probabilistic forecasts. A proper score rewards truthful probability reporting, and a strictly proper score makes the truthful report the unique optimum; Gneiting and Raftery [6] characterize regular proper scores as those generated by a convex function. In the binary case, proper scores admit threshold-mixture representations [7]; the beta family $S_{\alpha,\beta}$ uses this view to recover log and Brier scores as special cases while allowing bounded asymmetric cost shaping [8]. Calibration–resolution–uncertainty decompositions further show why marginal calibration is not enough: a forecast can be reliable on average while carrying little resolution about the outcome [9–11]. This distinction is central for agentic UQ, where an evaluator should reward informative prefix-conditioned probabilities rather than only binwise calibration. Our complete-data trajectory score applies this binary scoring-rule machinery at each prefix and aggregates the resulting scores with positive weights.

Proper scoring under censoring has been studied mainly in survival analysis, where follow-up can end before an event occurs, leaving its time unobserved and requiring observable-data extensions of complete-data scores. Rindt *et al.* [12] prove that a censored log-score extension is strictly proper for right-censored survival times, while Blanche *et al.* [13] show that common survival metrics, including C-index variants and integrated Brier-style scores, can fail propriety. Yanagisawa [14] systematizes this view by extending multiple proper scoring rules to censored observations and identifying when the required censoring weights are valid. Our censored extension borrows this conditional-projection

view, but targets a different forecast object: $F_t = \Pr(Y = 1 \mid \mathcal{H}_t)$, the probability of one fixed terminal outcome from the current agent prefix, rather than a cumulative event-time distribution. Thus the trace may rise or fall as evidence, mistakes, or recovery occur, unlike the monotone CDF target in standard survival analysis.

Agentic uncertainty quantification has so far focused mainly on producing uncertainty signals. Local predictors include verbalized confidence [15], token-level measures such as log-probability and predictive entropy [16, 17], and semantic entropy over sampled responses [18]. More recent agentic methods move from local signals to trajectory-level uncertainty: SAUP propagates step uncertainty with situation-aware weights [1], UProp studies how uncertainty is inherited across multi-step decision chains [2], AUQ evaluates trajectory confidence aggregators such as Φ_{last}, Φ_{avg}, and $\Phi_{\min}$ [3], and STeCa uses step-level calibration signals for agent learning [4]. These methods describe how uncertainty signals are produced, propagated, or used; we ask how to evaluate them once expressed as prefix-conditioned success probabilities.

Evaluation practice has not kept pace with the predictor side. UProp evaluates uncertainty with AUROC and selective-prediction metrics such as AUARC, while AUQ evaluates trajectory-level confidence aggregators with trajectory-ECE and trajectory-Brier. These metrics are useful diagnostics, but they target weaker objects: AUROC and AUPRC measure ranking, AUARC/AURC-style metrics measure selective-prediction ordering, ECE measures binwise marginal calibration of a scalar aggregate, and scalarized Brier is proper only for the aggregate confidence value supplied to it. Static-classification work has already shown that ECE is sensitive to binning and can obscure calibration structure [19, 20]. Work on selective prediction has also improved how answer-or-abstain policies are evaluated [21]. This is complementary to our goal: evaluating a fixed decision policy says how well that policy performs, not whether the probabilities behind it are truthful at every step. The problem is sharper for agents because most trajectory metrics first collapse the trace into a single number. Many different sequences of forecasts collapse to the same number, so even a proper score applied to that number cannot tell whether each intermediate forecast was truthful. Section 5.1 makes this precise (Theorems A and B; proofs in Appendix B).

3 Preliminaries

A scoring rule $S(p, y)$ rewards a reported probability p once the binary outcome y is observed; we use reward-oriented scores, so larger is better. Two standard examples are the log score $S_{\log}(p, y) = y \log p + (1 - y) \log(1 - p)$ and the Brier score $S_{\text{Brier}}(p, y) = -(p - y)^2$. A rule is proper if no report earns a higher expected score than the true probability, and strictly proper if the true probability is the unique maximizer, so a forecaster cannot gain by hedging toward 0.5 or by exaggerating. The log score penalizes confident errors without bound; Brier is bounded. TPS applies such a score at every step to forecasts of the same eventual outcome (Section 4).

An agent executes a task over up to T steps. At each step t, it observes o_t, takes action a_t, and emits a scalar forecast $F_t \in [0, 1]$. The prefix history is

$$h_t = (o_0, a_0, \ldots, o_t),$$

and $\mathcal{H}_t := \sigma(H_t)$ is the natural history filtration, $\mathcal{H}_1 \subseteq \cdots \subseteq \mathcal{H}_T$. A valid step-$t$ forecast is $\mathcal{H}_t$-measurable. This filtration view is consistent with the standard partially observable Markov decision process (POMDP) framing of agent interaction [22]: F_t can be viewed as the agent's belief state marginalized onto the binary success/failure partition. This is a prequential forecasting view [23]: forecasts are evaluated sequentially as information accumulates, but here each forecast targets the same eventual binary outcome rather than a new observation at each time step.

The forecast sequence is not a cumulative distribution function and need not be monotone: a calibrated agent may lower its forecast after a harmful action and raise it after acquiring useful evidence.

Fix an evaluation law $\mathbb{P}^\pi$ over full trajectories, induced by the benchmark distribution, environment randomness, tool randomness, and the fixed evaluation policy π. All conditional probabilities and expectations below are taken under this law. For readability, we suppress the superscript π and write $\mathbb{P}$ and $\mathbb{E}$ unless policy dependence is important. Let $Y \in \{0, 1\}$ denote the terminal task outcome. The truthful continuation-success process is

$$q_t := \mathbb{P}(Y{=}1 \mid \mathcal{H}_t) = \mathbb{E}[Y \mid \mathcal{H}_t]. \tag{1}$$

Equivalently, $(q_t)_{t=1}^T$ is the martingale of the terminal outcome with respect to the history filtration: its current value is the best conditional prediction of the final outcome given the current prefix. Operationally, q_t is the continuation-success frequency one would obtain by re-rolling many completions from the same prefix under the same evaluation law. It is not the agent's reported confidence; it is defined by $\mathbb{P}$. Under complete observation, Y is observed at termination and $Y_t = Y$ for all t: the target event does not change, only the information available about it.

For censored trajectories, let T^* denote the step at which the task outcome becomes determined and let C denote the exogenous termination step, such as a fixed step budget or compute budget. Let X denote the task description or instance, which we take to be included in the initial history H_0. We observe

$$Z = \min(T^*, C), \qquad \Delta = \mathbf{1}(T^* \le C).$$

We write z, δ for realized values of Z, Δ. If $\Delta = 1$, the trajectory is complete and Y is observed. If $\Delta = 0$, the trajectory is administratively censored at prefix Z and Y is unobserved.

Throughout, H_t denotes the random trajectory prefix at step t, while h_t denotes a realized history. The stopped history is denoted H_Z, and $F_{1:Z} := (F_1, \ldots, F_Z)$ denotes the forecast prefix observed before termination. We write $q_Z = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z)$ for the stopped random variable and q_z for its realized value on a trajectory censored at step z.

Assumption 1 (Non-informative censoring)**.**

$$T^* \perp C \mid X.$$

Assumption 1 requires that, conditional on the task instance, the external termination mechanism is independent of when the trajectory outcome would have been determined. This covers fixed step budgets and fixed compute budgets. It is violated by adaptive monitor-driven stopping, where a trajectory is stopped precisely because it appears likely to fail.

Assumption 2 (Administrative stop)**.**

$$Y \perp (C, \Delta) \mid \mathcal{H}_Z.$$

Assumption 2 requires that, once the stopped prefix is known, the fact that administrative censoring occurred at step Z carries no additional information about the eventual outcome. In other words, the observable prefix may be informative about success or failure, but the administrative stopping event itself is not.

Formally, S is proper if, conditionally on $\mathcal{H}_t$, $\mathbb{E}_{\mathbb{P}^\pi}[S(q_t, Y) \mid \mathcal{H}_t] \ge \mathbb{E}_{\mathbb{P}^\pi}[S(p, Y) \mid \mathcal{H}_t]$ for every $\mathcal{H}_t$-measurable p, and strictly proper if equality implies $p = q_t$ almost surely. We instantiate S with the beta family $S_{\alpha,\beta}$ from the Schervish–Buja threshold-mixture construction [7, 8], with $\nu(dc) = c^{\alpha-1}(1-c)^{\beta-1}\,dc$. It is strictly proper for all $\alpha, \beta > 0$, includes Brier (up to positive affine equivalence) and log as a limiting endpoint, and for $\alpha \ne \beta$ weights decision thresholds asymmetrically. Formulas, boundary behavior and proofs are in Appendix J.

4 Proper Scoring Rules for Agentic UQ

4.1 Trajectory score under complete observation

Let $w_{1:T}$ be a fixed evaluator-chosen weight schedule over trajectory steps, with $w_t > 0$ and, for reporting convenience, $\sum_{t=1}^T w_t = 1$. These weights are not uncertainty predictions; they encode which prefixes the evaluator wishes to emphasize. Uniform weights score every prefix equally, while front-loaded weights emphasize early prefixes, where miscalibration can shape later actions, observations, and opportunities for recovery [4, 24].

We define the *Trajectory Proper Score* (TPS) as the weighted lift of a strictly proper binary score to the prefix-conditioned forecast process:

$$\mathrm{TPS}(F_{1:T}, Y) = \sum_{t=1}^{T} w_t\, S_{\alpha,\beta}(F_t, Y). \tag{2}$$

The name is meant to emphasize that this is a family of trajectory-level proper scores rather than a single fixed scoring rule: any strictly proper binary score $S_{\alpha,\beta}$, combined with any fixed positive

weight schedule w_t, yields a strictly proper trajectory evaluator. We use subscripts to denote particular members, such as $\mathrm{TPS}_{\log}$, $\mathrm{TPS}_{\mathrm{Brier}}$, and $\mathrm{TPS}_{\beta(2,4)}$. Our experiments instantiate early-intervention weighting with the normalized linear front-loaded schedule

$$w_t = \frac{2(T-t+1)}{T(T+1)}.$$

For variable-length trajectories, T_i denotes the evaluated horizon of trajectory i: the realized terminal length for complete trajectories and the administrative budget/observed stop length for naturally censored max-step trajectories. The schedule w_{it} is constructed separately within each T_i and normalized over $t = 1, \ldots, T_i$; in artificial censoring, the original T_i-normalized weights are truncated at Z_i and are not renormalized over the observed prefix. Uniform weighting is included as a robustness check in the experiments. The strict-propriety result below does not depend on this particular schedule; it requires only that the scored steps and their weights are fixed in advance and strictly positive (Remark A.5).

Theorem 4.1 (Strict propriety under complete observation)**.** *If $S_{\alpha,\beta}$ is strictly proper and $w_t > 0$ for all t, then* TPS *is strictly proper for the prefix-conditioned success process:*

$$\mathbb{E}_{\mathbb{P}^\pi}[\mathrm{TPS}(q_{1:T}, Y)] \geq \mathbb{E}_{\mathbb{P}^\pi}[\mathrm{TPS}(F_{1:T}, Y)],$$

with equality if and only if $F_t = q_t$ a.s. for every t.

The proof is by conditional strict propriety at each filtration level and summing nonnegative regrets with positive weights; see Appendix A.

In our experiments we report log, Brier, and Beta(2,4) as representative members of the same proper family: log is the canonical unbounded default, Brier is the bounded symmetric baseline, and Beta(2,4) is a bounded asymmetric member. Appendix J also gives their cost-shaping interpretation.

4.2 Censoring as conditional projection

We now extend TPS to trajectories whose final outcome is not observed. Let

$$\mathcal{G}_Z := \sigma(H_Z, Z, \Delta, \Delta Y)$$

be the sigma-field generated by the stopped trajectory, the censoring indicator, and the observed terminal outcome on complete trajectories. The term ΔY records Y when $\Delta = 1$ and contributes no outcome information when $\Delta = 0$. If Y is already included in H_Z on complete trajectories, this term is redundant.

For a stopped prefix ending at Z, define the observable-prefix version of TPS by

$$\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z) := \sum_{t=1}^{Z} w_t \, S_{\alpha,\beta}(F_t, Y).$$

The weights w_t are inherited from the full trajectory score and are not renormalized over the observed prefix; this preserves comparability between complete-only and censored-prefix scores.

The abstract censored TPS is the conditional expectation of this observable-prefix complete-data score given the stopped trajectory:

$$\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(F_{1:Z}; \mathcal{G}_Z) := \mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z) \mid \mathcal{G}_Z\right]. \tag{3}$$

Theorem 4.2 (Abstract censored propriety on the observable prefix)**.** *If the per-step score $S_{\alpha,\beta}$ is proper under complete observation, then $\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}$ is proper for the observable prefix $F_{1:Z}$ under the stopped-data law. Strictness, when available, is restricted to the observable prefix.*

The proof is the conditional-expectation projection/tower-property argument; see Appendix C.

4.3 Exact reduced censored beta score

The preceding reduction yields a directly interpretable censored extension of TPS in the oracle case where the continuation-success weight q_Z is available. We write $\mathrm{TPS}^{\mathrm{cen,exact}}$ and $\mathrm{TPS}^{\mathrm{cen,simple}}$ for

censored extensions of TPS: the superscript indicates how the unobserved outcome is handled, while the subscript continues to indicate the binary score family. We define

$$\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}(F_{1:Z};\mathcal{G}_Z) = \sum_{t=1}^{Z} w_t \Big[\Delta\, S_{\alpha,\beta}(F_t, Y) + (1-\Delta)\Big(q_Z S_{\alpha,\beta}(F_t,1) + (1-q_Z) S_{\alpha,\beta}(F_t,0)\Big)\Big]. \tag{4}$$

Here $q_Z := \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z)$ is the stopped continuation-success process defined in Section 4.2.

Theorem 4.3 (Exact reduced censored beta score)**.** *Under Assumptions 1–2,* $\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}$ *is proper for every proper beta-family member. If* $S_{\alpha,\beta}$ *is strictly proper and* $w_t > 0$ *for all observed steps, then* $\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}$ *is strictly proper on the observable prefix* $F_{1:Z}$*.*

Appendix C proves the explicit reduced form, which reduces at the logarithmic endpoint to the corresponding soft-label log score on censored prefixes. Appendix H.2 validates this exact q_Z-weighted construction with Monte Carlo continuation rollouts.

4.4 Simple censored score for operational evaluation

The exact reduced score requires the continuation-success weight q_Z for censored prefixes. When this weight is not estimated, we use the pessimistic operational approximation

$$q_Z \approx 0,$$

which scores a censored prefix through the failure-side branch of the binary score. Equivalently, this sets the censored posterior success weight in (4) to zero:

$$\mathrm{TPS}^{\mathrm{cen,simple}}_{\alpha,\beta}(F_{1:Z};\mathcal{G}_Z) = \sum_{t=1}^{Z} w_t \left[\Delta\, S_{\alpha,\beta}(F_t, Y) + (1-\Delta)\, S_{\alpha,\beta}(F_t, 0)\right]. \tag{5}$$

At the logarithmic endpoint, the same failure-side reduction gives the simple censored-log score used as our primary operational censored evaluator:

$$\mathrm{TPS}^{\mathrm{cen,simple}}_{\log}(F_{1:Z};\mathcal{G}_Z) = \sum_{t=1}^{Z} w_t \left[\Delta\{Y \log F_t + (1-Y)\log(1-F_t)\} + (1-\Delta)\log(1-F_t)\right]. \tag{6}$$

When interpreted as a proper score, $\mathrm{TPS}^{\mathrm{cen,simple}}$ elicits the pseudo-label target

$$m_t = \mathbb{E}_{\mathbb{P}^\pi}[\Delta Y \mid \mathcal{H}_t],$$

not the original continuation-success target $q_t = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_t)$, unless the missing success mass is zero. We therefore use $\mathrm{TPS}^{\mathrm{cen,simple}}$ only as an explicit pessimistic $q_Z \approx 0$ approximation to the exact reduced score. The logarithmic instance is our primary operational censored evaluator on WebShop; Brier and Beta-family instances are reported only as robustness checks. Appendix C gives the pseudo-label target result and proof.

5 Evaluation Metrics for Agentic UQ

5.1 Why common trajectory evaluators are insufficient

We now organize common agentic UQ evaluators by the forecast object they elicit: a rank ordering, a binwise-calibrated scalar, a collapsed probability, or the full prefix-conditioned probability process. Formally, Theorem A shows that T-ECE can tie non-truthful resolution-losing forecasts with the truthful process, while Theorem B shows that scalarized proper losses elicit only $C = \Phi(F_{1:T})$, not the full trace.

AUROC and rank metrics. AUROC evaluates whether a scalar trajectory score ranks successful and unsuccessful trajectories correctly. It is invariant under any strictly monotone reparametrization of the scalar score $C = \Phi(F_{1:T})$ supplied to it [25]. Thus AUROC can distinguish useful ordering

from useless ordering, but it cannot distinguish well-scaled probabilities from overconfident or underconfident monotone distortions. The same issue applies to other rank- or threshold-based diagnostics such as AUPRC and risk–coverage/AURC: they are useful discrimination diagnostics, but they do not elicit the probability trace.

Trajectory ECE. Trajectory ECE [3] collapses the forecast trace to a scalar $C = \Phi(F_{1:T})$ and checks whether empirical success rates match average confidence within fixed bins. A forecast that is right on average within every bin scores zero T-ECE, regardless of whether it separates successful from failed trajectories within those bins. This distinction matters: a proper forecast must not only be calibrated on average (*Reliability*) but also concentrate probability mass where outcomes differ (*Resolution*). T-ECE measures only the first component of this calibration–resolution–uncertainty decomposition [9–11] and is blind to the second. Concretely, the truthful process q and its binwise average $G = \mathbb{E}_{\mathbb{P}^\pi}[q \mid \mathrm{bin}(q)]$ can both achieve T-ECE$= 0$, even though G discards all within-bin discriminative information.

Trajectory Brier and scalarized proper scores. Brier and log loss are strictly proper for a single binary probability forecast. When a trajectory is first collapsed to $C = \Phi(F_{1:T})$ and the proper score is applied only to C, however, the resulting scalarized score elicits the collapsed scalar, not the full trace. Common AUQ-style aggregators, including Φ_{last}, Φ_{avg}, $\Phi_{\min}$, and weighted averages, can therefore hide or reward non-truthful intermediate forecasts. The scalarization result is sharper: Φ_{last}-Brier is proper but not strictly proper for the trace, while Φ_{avg}-, $\Phi_{\min}$-, and Φ_w-Brier can make the truthful trace strictly suboptimal.

Each evaluator above is well suited to what it measures: AUROC, AUPRC, and AURC for rank discrimination; T-ECE for binwise marginal calibration; and scalarized Brier/log scores for a collapsed scalar probability. The gap is specific: none of them strictly elicits the full prefix-conditioned success-probability process (q_t). TPS fills this gap under complete observation; $\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}$ extends it to the observable prefix under administrative censoring.

5.2 Reporting convention

TPS should be the primary metric whenever F_t is interpreted as a probability of eventual task success. AUROC/AUPRC remain useful for ranking, risk–coverage for a specified selective policy, and ECE as a bin-dependent diagnostic, but none replaces a proper probability-quality score, and none scores censored traces natively: they must first be discarded, continued, or mislabeled. Complete TPS takes $O(T)$ time and constant additional memory, with no bins or learned evaluator. Section 7 turns this convention into a step-by-step evaluation protocol.

6 Experiments

The preceding sections show that common trajectory diagnostics and TPS answer different evaluation questions. We now ask whether this difference is large enough to change empirical verdicts on real agent traces. We use these predictor streams to probe evaluator behavior, not to rank uncertainty methods.

Setup. Experiments use Gemma 4 31B in a fixed ReAct harness on Tau2-Bench, StrategyQA, HotpotQA, and WebShop. We evaluate five per-step probability streams: verbal confidence, completion-token probability, completion-entropy confidence, action-span token probability, and action-span entropy confidence, plus a base-rate reference. Primary analyses use log score with linear-front weights and cross-fitted Platt calibration where reported. Predictor formulas and transparency tables are in Appendix F; calibration is in Appendix D; preprocessing, prompts, and hyperparameters are in Appendix K; and robustness sweeps are in Appendix I.

6.1 Calibration-invariance gap

We test whether rank-based evaluation and proper trajectory scoring remain aligned when a forecast stream is recalibrated. We take the model's verbal confidences on Tau2-Bench, which are heavily saturated near $\{0.90, 0.95, 1.00\}$, and compare the raw stream with a Platt-recalibrated version. The base-rate stream $F_t \equiv \bar{y}$ serves as an uninformative reference. If AUROC and $\mathrm{TPS}_{\log}$ (the log

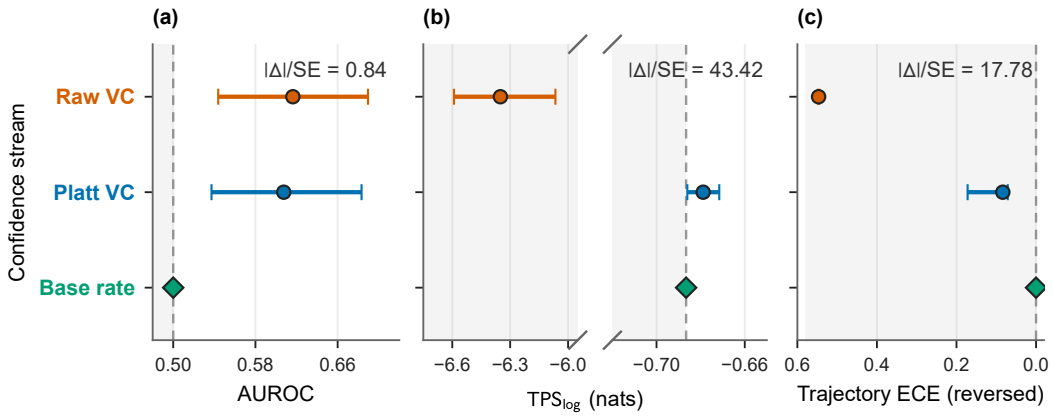


Figure 1: Tau2-Bench calibration gap ($n = 201$). Raw verbal confidence (orange), Platt-recalibrated confidence (blue), and the base-rate reference (green). Whiskers are 95% intervals from 1,000 paired task-bootstrap resamples; annotations give $|\Delta|$ divided by its paired bootstrap SE. Shaded regions are worse than base rate. Right is better; T-ECE is reversed and $\mathrm{TPS}_{\log}$ uses a broken axis.

endpoint of TPS, i.e. $\alpha = \beta = 0$) led to the same practical verdict, recalibration should not separate them by many bootstrap standard errors.

Figure 1 shows the opposite. Let Δ/SE denote the raw-to-Platt change divided by its paired bootstrap standard error. Recalibration changes AUROC by only -0.009 (95% CI $[-0.029, +0.012]$), less than one paired-bootstrap SE, while improving $\mathrm{TPS}_{\log}$ by 5.67 nats ($\Delta/\mathrm{SE} \approx 43$). The corresponding AUPRC and AURC changes are also small relative to the $\mathrm{TPS}_{\log}$ shift. Under $\mathrm{TPS}_{\log}$, the raw stream lies 5.66 nats below the base-rate reference; the Platt stream lies 0.008 nats above it. Appendix E repeats the comparison with the rank-metric input held fixed by construction; AUROC/AUPRC/AURC are identical across transforms while $\mathrm{TPS}_{\log}$ spans 5.7 nats. The result does not depend on Platt scaling: isotonic and beta calibration, fit on the same folds, give the same 5.67-nat improvement, and their AUROC changes are again indistinguishable from zero (Appendix D.1).

After calibration, verbal confidence barely exceeds the base-rate reference, improving probability scale but not discrimination. AUROC treats the raw and recalibrated streams as nearly equivalent because their rankings are nearly equivalent, while $\mathrm{TPS}_{\log}$ detects that one stream is badly mis-scaled as a probability forecast. This distinction matters whenever F_t is consumed as a probability for deferral, thresholding, human handoff, or combination with a cost model.

6.2 Controlled censoring validation

Before applying the censored estimator to natural truncation, we validate it under artificial censoring on complete StrategyQA, Tau2-Bench, and HotpotQA trajectories. Appendix I shows that the closed-form prefix-swap/tail-omission decomposition matches the directly computed $\mathrm{TPS}^{\mathrm{cen,simple}} - \mathrm{TPS}^{\mathrm{complete}}$ difference to numerical precision across score families, weight schedules, censoring rates, and datasets. The validation also predicts the sign regime used to interpret WebShop: low-confidence censored prefixes receive mild failure-branch corrections, while overconfident censored prefixes are penalized sharply.

6.3 Natural censoring on WebShop

We next move from artificial censoring to naturally truncated agent traces. WebShop is the relevant case: many trajectories hit the benchmark step budget before an outcome is observed. Of 500 trajectories, 192 parse-error truncations are excluded as informative censoring. These parse failures arise from the agent's own malformed output and plausibly correlate with task success, so treating them as administrative censoring would violate the non-informative censoring assumption. The working sample contains $n = 308$ trajectories: 163 completed trajectories with observed outcomes ($\delta = 1$) and 145 administrative max-step stops ($\delta = 0$), for a 47.08% administrative-censoring rate.

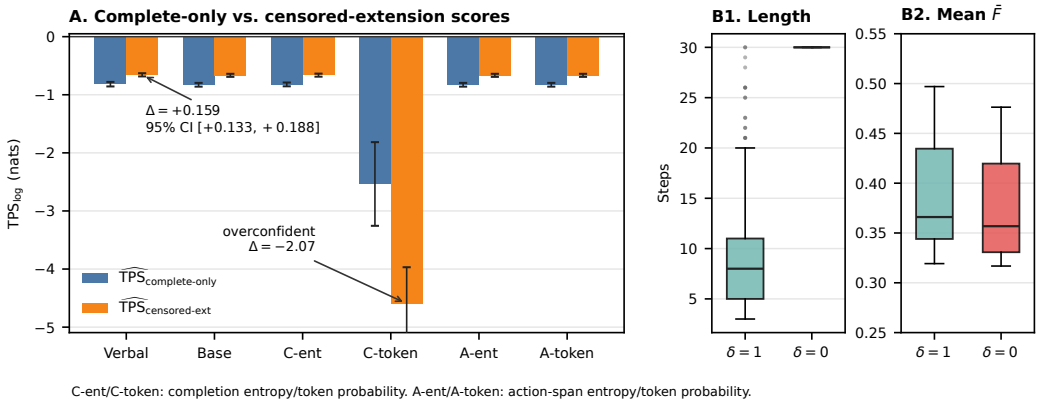


Figure 2: WebShop natural censoring. Panel A compares complete-only and simple-censored scores; the primary verbal-confidence score shifts by $+0.159$ nats with 95% CI $[+0.133, +0.188]$. Panel B shows the selection diagnostic: admin-censored trajectories are longer, while verbal confidence barely separates completed and censored subsets. Whiskers are 95% bootstrap intervals.

The censored subset is structurally different from the completed subset. Admin-censored trajectories are $3\times$ longer on average than completed ones (30.0 vs. 9.88 observed steps), consistent with harder tasks exhausting the step budget. Yet verbal confidence does not materially separate the two subsets ($\Delta\bar{F} = -0.007$): the predictor is nearly flat along a difficulty axis visible in the stopping pattern. Complete-only evaluation therefore discards a large, structurally harder subset of the benchmark. Figure 2 shows this selection diagnostic alongside the primary score shift. We report

$$\Delta_{\text{practice}} = \widehat{\text{TPS}}_{\text{censored-ext}} - \widehat{\text{TPS}}_{\text{complete-only}},$$

with positive values meaning that censored-aware evaluation gives a higher average score than complete-only evaluation.

Our main reported result uses Platt-calibrated verbal confidence scored with the log rule and linear-front weights. Complete-only evaluation uses only the 163 completed trajectories and gives $\widehat{\text{TPS}}_{\text{complete-only}} = -0.816$ nats; the simple censored extension also includes the 145 max-step prefixes and gives $\widehat{\text{TPS}}_{\text{censored-ext}} = -0.657$ nats. The paired shift is $+0.159$ nats, with a 95% bootstrap CI of $[+0.133, +0.188]$. This is an evaluator effect, not an improvement in task performance: the additional trajectories have unobserved outcomes and are scored under the failure-side $q_z \approx 0$ approximation. Normalized by the 47.08% censoring rate, the shift is $+0.337$ nats per unit censoring rate, close to the corresponding artificial-censoring value on Tau2 ($\approx +0.404$ nats; Table 11, Appendix I.1).

Across the score-family and weight-schedule sweep, the sign pattern is stable: all non-reference predictor rows are positive except completion token probability, the only predictor with $\bar{F} > 0.5$. Under log score with linear-front weights, this predictor receives a -2.07 nat shift. The pattern is consistent with the artificial-censoring analysis in Section 6.2: low-confidence censored prefixes receive a mild failure-branch correction, while overconfident censored prefixes are penalized sharply. Thus censored-aware scoring changes not only the average score, but also how different confidence regimes are treated. Appendix H analyzes q_Z estimation sensitivity and endpoint intervals on the resumable HotpotQA audit; Appendix M reports sample-size stability, including the WebShop shift under the declared $q_Z \approx 0$ protocol.

7 Practitioner Guide

The results of Sections 4–6 suggest six steps for evaluating an agent's per-step uncertainty. Steps 2–4 should be decided before comparing systems. Appendix O explains each step and works through an example, and Appendix N provides a reporting template.

1. **Forecast object.** TPS scores a probability of eventual success at every step. Convert raw signals such as token probabilities into probabilities using held-out data (Appendix D). Single-number trajectory summaries are not per-step forecasts and fall outside TPS.
2. **Score.** By default, use log TPS as the main score and Brier as a bounded check. If the deployment acts at a known threshold τ, also report a beta score centered at τ (Appendix J.3). Choose all scores in advance; never report only the most favorable one.
3. **Weights.** Any fixed positive weights keep TPS proper; they only decide which steps count most. Use uniform weights unless the task calls for emphasis, such as early steps when early intervention matters. Censored prefixes keep their original weights.
4. **Stops.** Label each trajectory as completed, stopped by a fixed budget, or stopped for an informative reason such as a parse failure, and report the counts. Exclude informative stops from censored scoring (Appendix G).
5. **Censored prefixes.** If stopped prefixes can be resumed, estimate q_Z by rerunning them and report exact TPS. Otherwise use a separately trained estimate of q_Z that meets Remark C.4, or report the $q_Z \approx 0$ score together with its range over $q_Z \in [0, 1]$ (Remark C.5, Appendix H).
6. **Interpretation.** Report TPS as the main score, with rank and calibration metrics as complements and the base-rate reference as a baseline, and state what the intervals resample. TPS evaluates recorded trajectories under a fixed policy (Section 8).

8 Conclusion and Limitations

Evaluating agentic uncertainty requires scoring a probability trace, not only summarizing it. Existing trajectory evaluators, including AUROC, T-ECE, and scalarized T-Brier under common aggregators, elicit rank orderings, binwise calibration, or collapsed scalars. None strictly elicits the prefix-conditioned success process (q_t) that a calibrated agent should report. TPS, built from any strictly proper binary score and positive trajectory weights, provides a trajectory-level evaluator that does.

Censoring further changes the evaluation population. When benchmarks impose step budgets, complete-only evaluation silently discards truncated trajectories. On WebShop, the administratively censored subset is structurally different from the completed subset, and censored-aware scoring changes the verdict by 0.159 nats with a 95% confidence interval excluding zero. Together with the Tau2-Bench recalibration result, where $\mathrm{TPS}_{\log}$ moves by $\Delta/\mathrm{SE} \approx 43$ while AUROC moves by less than one paired-bootstrap SE, these results show that evaluator choice is not cosmetic when F_t is consumed as a probability for deferral, thresholding, reflection, or human handoff.

Limitations. Our experiments score recorded trajectories under a fixed policy. They show which confidence stream is more truthful, but not whether acting on better-calibrated confidence would improve task success; that question needs a closed-loop experiment. The censored extension assumes that stops are administrative (Assumptions 1–2). Stops triggered by the agent's own confidence, truncated or summarized histories, and paused and resumed episodes each need their own analysis. The complete-observation score extends naturally beyond binary outcomes, for example to graded rewards or several outcome categories, but extending the censored score requires more than the single number q_Z (Remark A.4). Propriety also holds only for the distribution being evaluated: if TPS is used as a training loss, it should be checked on held-out tasks and under distribution shift. TPS measures whether probabilities are truthful, not whether the agent is capable, and a good score alone does not justify reducing human oversight (Appendix J.4). Finally, we use one model and one ReAct-style harness. Because rank metrics ignore probability scale by construction (Appendix E), the gap does not depend on a particular model, but its size does; broader validation remains future work.

References

[1] Qiwei Zhao, Xujiang Zhao, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, and Haifeng Chen. Saup: Situation awareness uncertainty propagation on llm agent, 2024. URL `https://arxiv.org/abs/2412.01033`.

[2] Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. Uprop: Investigating the uncertainty propagation of llms in multi-step agentic decision-making, 2025. URL `https://arxiv.org/abs/2506.17419`.

[3] Jiaxin Zhang, Prafulla Kumar Choubey, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Agentic uncertainty quantification, 2026. URL https://arxiv.org/abs/2601.15703.

[4] Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. Steca: Step-level trajectory calibration for llm agent learning, 2025. URL https://arxiv.org/abs/2502.14276.

[5] Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large-language model agents, 2025. URL https://arxiv.org/abs/2505.22655.

[6] Tilmann Gneiting and Adrian Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 03 2007. doi: 10.1198/016214506000001437.

[7] Mark J. Schervish. A general method for comparing probability assessors. *The Annals of Statistics*, 17(4):1856–1879, 1989. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2241668.

[8] Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. 01 2005.

[9] Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12:595–600, 1973. URL https://api.semanticscholar.org/CorpusID:121053719.

[10] Morris H. Degroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983. URL https://api.semanticscholar.org/CorpusID:109884250.

[11] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009. ISSN 1477-870X. doi: 10.1002/qj.456. URL http://dx.doi.org/10.1002/qj.456.

[12] David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. Survival regression with proper scoring rules and monotonic neural networks. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 1190–1205. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/rindt22a.html.

[13] Paul Blanche, Michael W Kattan, and Thomas A Gerds. The c-index is not proper for the evaluation of t-year predicted risks. *Biostatistics*, 20(2):347–357, 04 2019. ISSN 1465-4644. doi: 10.1093/biostatistics/kxy006. URL https://doi.org/10.1093/biostatistics/kxy006.

[14] Hiroki Yanagisawa. Proper scoring rules for survival analysis, 2023. URL https://arxiv.org/abs/2305.00621.

[15] Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent, 2024. URL https://arxiv.org/abs/2401.14016.

[16] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction, 2021. URL https://arxiv.org/abs/2002.07650.

[17] Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models, 2024. URL https://arxiv.org/abs/2307.01379.

[18] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

[19] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration, 2020. URL https://arxiv.org/abs/1909.10155.

[20] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B. Schön. Evaluating model calibration in classification, 2019. URL https://arxiv.org/abs/1902.06977.

[21] Jeremias Traub, Till J. Bungert, Carsten T. Lüth, Michael Baumgartner, Klaus H. Maier-Hein, Lena Maier-Hein, and Paul F Jaeger. Overcoming common flaws in the evaluation of selective classification systems, 2024. URL https://arxiv.org/abs/2407.01032.

[22] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

[23] A. P. Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. *Royal Statistical Society. Journal. Series A: General*, 147(2): 278–290, 03 1984. ISSN 0035-9238. doi: 10.2307/2981683. URL https://doi.org/10.2307/2981683.

[24] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality, 2023. URL https://arxiv.org/abs/2305.18654.

[25] James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143 1:29–36, 1982. URL https://api.semanticscholar.org/CorpusID:10511727.

[26] R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. doi: 10.1109/TNN.1998.712192.

[27] Heejung Bang and James M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 12 2005. ISSN 0006-341X. doi: 10.1111/j.1541-0420.2005.00377.x. URL https://doi.org/10.1111/j.1541-0420.2005.00377.x.

[28] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 02 2018. ISSN 1368-4221. doi: 10.1111/ectj.12097. URL https://doi.org/10.1111/ectj.12097.

[29] Google DeepMind. Gemma 4 31B IT. https://huggingface.co/google/gemma-4-31B-it, 2026. Hugging Face model card.

[30] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

[31] Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

[32] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies, 2021. URL https://arxiv.org/abs/2101.02235.

[33] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

[34] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

Appendix

Reading guide. Appendix A proves the complete-observation result in Section 4.1; Appendix B supports the evaluator limitations in Section 5.1; and Appendix C proves the censored reductions in Sections 4.2–4.4, including plug-in error propagation. Appendix D documents calibration and its sensitivity analysis for Section 6.1; Appendix E isolates the fixed-rank gap; and Appendix F defines the predictor streams used in Section 6. Appendix G supports the stopping classification in Section 6.3, while Appendix H audits the exact q_Z correction and continuation budgets. Appendix I gives score/weight sensitivity and Appendix M gives sample-size stability for Section 6. Appendix J develops the score-family, cost-shaping, and calibration-loss interpretations used in Sections 3 and 7. Appendix K records experimental settings and task grouping; Appendix L records asset licenses; and Appendix N provides the reporting template for Section 7. Appendix O expands Section 7 with rationale and a worked example.

A Proof of Theorem 4.1 (Complete Observation)

We give the full proof of Theorem 4.1 and record several remarks on the filtration argument, the prequential interpretation, the score family, non-binary outcomes, and the fixed-weight scope.

Proof. Let $F_{1:T}$ be any adapted forecast sequence with F_t $\mathcal{H}_t$-measurable, and let

$$q_t := \mathbb{P}^\pi(Y{=}1 \mid \mathcal{H}_t).$$

The trajectory weights $w_{1:T}$ are fixed by the evaluator and satisfy $w_t > 0$ for every t. By linearity of expectation,

$$\mathbb{E}_{\mathbb{P}^\pi}[\mathrm{TPS}(F_{1:T}, Y)] = \sum_{t=1}^{T} w_t \, \mathbb{E}_{\mathbb{P}^\pi}[S_{\alpha,\beta}(F_t, Y)]. \tag{7}$$

Fix a step t. Since F_t is $\mathcal{H}_t$-measurable and $Y \mid \mathcal{H}_t$ has Bernoulli parameter q_t, conditional strict propriety of the binary score gives

$$R_t := \mathbb{E}_{\mathbb{P}^\pi}[S_{\alpha,\beta}(q_t, Y) - S_{\alpha,\beta}(F_t, Y) \mid \mathcal{H}_t] \geq 0 \quad \text{a.s.}, \tag{8}$$

with equality if and only if $F_t = q_t$ a.s.

Multiplying (8) by $w_t > 0$, taking expectations, and summing over t yields

$$\sum_{t=1}^{T} w_t \, \mathbb{E}_{\mathbb{P}^\pi}[R_t] = \mathbb{E}_{\mathbb{P}^\pi}[\mathrm{TPS}(q_{1:T}, Y)] - \mathbb{E}_{\mathbb{P}^\pi}[\mathrm{TPS}(F_{1:T}, Y)] \geq 0. \tag{9}$$

Thus the truthful process $q_{1:T}$ maximizes expected trajectory score.

It remains to show uniqueness. If equality holds in (9), then each term in the nonnegative weighted sum must be zero: $\mathbb{E}_{\mathbb{P}^\pi}[R_t] = 0$ for every t. Since $R_t \geq 0$ a.s., this implies $R_t = 0$ a.s. for every t. By the equality condition in the per-step strict propriety statement, $F_t = q_t$ a.s. for every t. Hence $q_{1:T}$ is the unique maximizer of expected trajectory score, up to almost-sure equivalence. □

Remark A.1 (Step dependence does not affect the proof). The forecasts $(F_1, \ldots, F_T)$ may be arbitrarily dependent across steps, and every step may target the same terminal outcome Y. The proof does not require independent step labels or independent forecast errors. The key point is conditional: at each filtration level $\mathcal{H}_t$, the reported forecast F_t is measurable with respect to the information available at that step, and the conditional law of Y is Bernoulli with parameter q_t. Strict propriety is therefore applied separately inside each conditional expectation, and the resulting nonnegative regrets are then aggregated with positive weights.

Remark A.2 (Prequential interpretation). The construction is prequential in spirit: the evaluator scores a sequence of probability forecasts as information accumulates along the interaction history. It differs from classical one-step-ahead prequential forecasting because all forecasts target the same terminal binary outcome Y, rather than a sequence of distinct future observations. Accordingly,

$$q_t = \mathbb{E}_{\mathbb{P}^\pi}[Y \mid \mathcal{H}_t]$$

is the truthful success probability given the information available at step t. As the history grows, this conditional probability can increase or decrease depending on what the agent observes or does. This is why no monotonicity assumption is needed. A truthful success forecast may rise after useful evidence or successful tool use, and may fall after a harmful action or failed observation.

Remark A.3 (Role of the binary scoring-rule family). Theorem 4.1 assumes that the per-step binary score $S_{\alpha,\beta}$ is strictly proper. That property is established for the beta family in Appendix J. The proof above uses only the consequence of that result: for any $\mathcal{H}_t$-measurable forecast F_t, the conditional expected score is uniquely maximized by reporting $q_t = \mathbb{P}^\pi(Y{=}1 \mid \mathcal{H}_t)$.

Gneiting–Raftery gives the convex-function characterization of proper scores; the binary specialization yields the Savage form; and the Schervish threshold-mixture representation gives a convenient route to strict propriety for beta-family scores whose mixing measure has full support on $(0, 1)$. The logarithmic endpoint is handled as the usual strictly proper binary log score.

Remark A.4 (Beyond binary outcomes). Nothing in the proof above requires a binary outcome: it only uses a strictly proper score at each step and fixed positive weights. The same construction therefore applies to richer outcomes once a suitable per-step score is chosen. A graded reward can be scored through its expected value or its full distribution, a categorical outcome through a multiclass proper score, and several objectives through a joint distribution or a utility function fixed in advance. Each choice determines what the agent is rewarded for reporting. Only two parts of this paper are specific to binary outcomes: the beta family of scores, and the censored score, which summarizes the missing outcome by the single probability q_Z. For richer outcomes, the censored score would need the expected value or full distribution of the missing outcome instead; working this out is left to future work.

Remark A.5 (Scope of the fixed-weight statement). The theorem is stated for a fixed evaluated horizon and fixed positive evaluator weights. Length-specific reporting conventions, such as using a normalized linear-front schedule separately for each observed complete trajectory length, should be read as applying the same theorem within each fixed length convention. Stopped or censored prefixes are handled by the separate conditional-projection construction in Appendix C. Because the scored steps are fixed in advance, TPS evaluates the trajectory the agent actually produced and is not meant to be invariant to edits of it. Reordering actions gives a different trajectory that the agent would have to re-run, and duplicating or adding scored steps changes the evaluation protocol itself. Steps that are not scored do not affect the score.

B Proofs of Theorems A and B

The results in this appendix should not be read as dismissing rank metrics, ECE, or scalarized Brier scores. AUROC, AUPRC, and risk–coverage evaluate ranking; ECE diagnoses binwise marginal calibration; and proper binary losses such as Brier and log loss are proper for a single scalar forecast. The negative results below show only that these evaluator families do not strictly elicit the full prefix-conditioned forecast process (q_t).

B.1 Theorem A: T-ECE is Resolution-blind

Let $C = \Phi(F_{1:T}) \in [0, 1]$ be a scalar trajectory confidence obtained from a forecast trace by an aggregator Φ, and let $\mathcal{B} = \{B_1, \ldots, B_K\}$ be a fixed partition of $[0, 1]$. The population trajectory ECE is

$$\text{T-ECE}_{\mathcal{B}}(C) = \sum_{k=1}^{K} \mathbb{P}^\pi(C \in B_k)\, \left|\mathbb{E}_{\mathbb{P}^\pi}[Y \mid C \in B_k] - \mathbb{E}_{\mathbb{P}^\pi}[C \mid C \in B_k]\right|. \tag{10}$$

This is a calibration functional, not a single-observation scoring rule.

The Reliability–Resolution–Uncertainty decomposition [9–11] states that, for a negatively oriented proper loss L and scalar forecast C,

$$\mathbb{E}_{\mathbb{P}^\pi}[L(C, Y)] = \underbrace{\mathbb{E}_{\mathbb{P}^\pi}\, d(C, \mathbb{E}_{\mathbb{P}^\pi}[Y \mid C])}_{\text{Reliability}} - \underbrace{\mathbb{E}_{\mathbb{P}^\pi}\, d\big(\mathbb{E}_{\mathbb{P}^\pi}[Y \mid C], \bar{Y}\big)}_{\text{Resolution}} + \underbrace{d(\bar{Y}, Y)}_{\text{Uncertainty}}, \tag{11}$$

where d is the associated Bregman divergence and $\bar{Y} = \mathbb{E}_{\mathbb{P}^\pi}[Y]$. T-ECE evaluates the Reliability component only: it asks whether empirical success frequencies match average confidence within

bins. It does not penalize loss of Resolution, the component that distinguishes an informative truthful forecast from an uninformative but calibrated one.

Theorem A (T-ECE is resolution-blind)**.** *There exists a data-generating distribution and a non-truthful forecast process $G \neq q$ almost surely such that*

$$\text{T-ECE}_{\mathcal{B}}(\Phi(G_{1:T})) \leq \text{T-ECE}_{\mathcal{B}}(\Phi(q_{1:T})).$$

Moreover, binwise marginalization of the truthful forecast can preserve zero binwise calibration error while strictly reducing Resolution whenever the truthful conditional probability varies within bins.

Proof. It suffices to take $T = 1$. Let $H \in \{a, b\}$, with $\mathbb{P}^\pi(H = a) = \mathbb{P}^\pi(H = b) = \frac{1}{2}$, and let

$$\mathbb{P}^\pi(Y = 1 \mid H = a) = 0.2, \qquad \mathbb{P}^\pi(Y = 1 \mid H = b) = 0.8.$$

The truthful forecast is $q(H) = \mathbb{P}^\pi(Y = 1 \mid H)$. Since $\mathbb{E}_{\mathbb{P}^\pi}[Y \mid q = p] = p$, the truthful forecast achieves $\text{T-ECE}(q) = 0$ for any binning that separates or contains these forecast values without inducing binwise miscalibration.

Now define the constant forecast $G(H) \equiv 0.5$. Since the marginal success rate is

$$\mathbb{E}_{\mathbb{P}^\pi}[Y] = \tfrac{1}{2}(0.2) + \tfrac{1}{2}(0.8) = 0.5,$$

we have $\mathbb{E}_{\mathbb{P}^\pi}[Y \mid G = 0.5] = 0.5$, and hence $\text{T-ECE}(G) = 0$. Thus G ties the truthful forecast under T-ECE despite $G \neq q$ almost surely.

The difference is Resolution. Under the Brier divergence, the constant forecast has zero Resolution, while the truthful forecast achieves

$$\tfrac{1}{2}(0.2 - 0.5)^2 + \tfrac{1}{2}(0.8 - 0.5)^2 = 0.09 > 0.$$

T-ECE is blind to this distinction.

The same phenomenon holds more generally for forecasts that replace q by its conditional average on the cells of a fixed bin map. Such forecasts preserve binwise marginal calibration by construction, but remove within-bin variation. Whenever q has positive within-bin variance, this averaging strictly reduces the Resolution term in (11), while the T-ECE functional in (10) does not penalize that loss. Therefore T-ECE is not strictly proper for the full prefix-conditioned forecast process. □

B.2 Theorem B: Scalarized proper losses are not strictly proper for the trace

Let $L(p, Y)$ be a negatively oriented strictly proper binary loss, so lower expected loss is better. Given an aggregator Φ, define the scalarized trajectory loss

$$\mathcal{L}_\Phi(F_{1:T}, Y) := L(\Phi(F_{1:T}), Y)\,. \tag{12}$$

This includes scalarized trajectory Brier as the special case $L(p, Y) = (p - Y)^2$. The issue is not that Brier or log loss are improper for a single binary probability; they are proper for that scalar. The issue is that applying them after a deterministic collapse $\Phi(F_{1:T})$ changes the elicited object from the trace (F_t) to the scalar $C = \Phi(F_{1:T})$.

Write

$$q_T^* := \mathbb{E}_{\mathbb{P}^\pi}[Y \mid \mathcal{H}_T]$$

for the conditionally optimal scalar input to a strictly proper binary loss given the final observed history. In the two-step examples below, $q_T^* = q_2$.

Theorem B (Scalarized proper losses elicit the collapsed scalar, not the trace)**.** *Let L be a strictly proper binary loss and let Φ be one of the AUQ-style aggregators Φ_{last}, Φ_{avg}, $\Phi_{\min}$, or a linear aggregator*

$$\Phi_w(F_{1:T}) = \sum_{t=1}^{T} w_t F_t, \qquad w_t \geq 0, \qquad \sum_{t=1}^{T} w_t = 1,$$

with at least two positive weights.

(i) Non-uniqueness. *For each of these aggregators, there exists a data-generating distribution and an adapted non-truthful trace $G_{1:T} \neq q_{1:T}$ such that*

$$\mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_\Phi(G_{1:T}, Y)] = \mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_\Phi(q_{1:T}, Y)].$$

Hence the scalarized loss is not strictly proper for the full trace.

(ii) Strict suboptimality. *For Φ_{avg}, $\Phi_{\min}$, and any Φ_w with $T \geq 2$ and at least two positive weights, there are data-generating distributions for which the truthful aggregate $\Phi(q_{1:T})$ differs from the conditionally optimal scalar q_T^* on a positive-probability event, and an adapted non-truthful trace $G_{1:T}$ can achieve that scalar optimum. In such cases,*

$$\mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_\Phi(G_{1:T}, Y)] < \mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_\Phi(q_{1:T}, Y)].$$

For Φ_{last}, part (i) holds but part (ii) does not: the truthful trace is optimal, but not uniquely optimal, because the prefix is invisible to the loss.

We prove the aggregator-specific cases below.

Corollary B1: Φ_{last} is proper for the last scalar, not strictly proper for the trace

Let

$$\Phi_{\mathrm{last}}(F_{1:T}) = F_T.$$

Then $\mathcal{L}_{\Phi_{\mathrm{last}}}(F, Y) = L(F_T, Y)$ depends only on the final forecast. By strict propriety of L, the expected loss is minimized by $F_T = q_T$. However, any adapted trace $G_{1:T}$ satisfying $G_T = q_T$ and $G_t \neq q_t$ on a positive-probability set for some $t < T$ achieves the same expected loss as the truthful trace. Thus Φ_{last}-Brier, and likewise any last-step scalarized proper loss, is proper for the collapsed final scalar but not strictly proper for the full trace.

Corollary B2: Φ_{avg} can make the truthful trace strictly suboptimal

Let $T = 2$. Let $q_1 = 0.5$, and let

$$q_2 \in \{0.7, 0.3\}$$

with each value occurring with probability $1/2$, where $\mathbb{E}_{\mathbb{P}^\pi}[q_2 \mid \mathcal{H}_1] = q_1 = 0.5$ by the law of iterated expectations. This is a valid prefix-conditioned success process.

For the truthful trace,

$$\Phi_{\mathrm{avg}}(q) = \frac{q_1 + q_2}{2} \in \{0.6, 0.4\}.$$

But, conditional on $\mathcal{H}_2$, the strictly proper scalar loss is minimized by reporting $q_T^* = q_2$, not $(q_1 + q_2)/2$.

Define an adapted non-truthful trace

$$G_1 = 0.5, \qquad G_2 = 2q_2 - 0.5 \in \{0.9, 0.1\}.$$

Then

$$\Phi_{\mathrm{avg}}(G) = \frac{G_1 + G_2}{2} = q_2.$$

Thus G achieves the conditionally optimal scalar $q_T^* = q_2$, while the truthful trace does not. For the Brier loss, the truthful aggregate is off by 0.1 on every realization, so it incurs an excess conditional loss of $0.1^2 = 0.01$. Hence the truthful trace is strictly suboptimal under Φ_{avg}-Brier.

Corollary B3: $\Phi_{\min}$ can make the truthful trace strictly suboptimal

Use the same two-step process as above:

$$q_1 = 0.5, \qquad q_2 \in \{0.7, 0.3\}.$$

For the truthful trace,

$$\Phi_{\min}(q) = \min(q_1, q_2) \in \{0.5, 0.3\}.$$

Define an adapted non-truthful trace

$$G_1 = 1, \qquad G_2 = q_2.$$

Then

$$\Phi_{\min}(G) = q_2.$$

On the event $q_2 = 0.7$, the truthful scalar is 0.5, while the non-truthful scalar is the conditionally optimal value $q_T^* = q_2 = 0.7$. Computing the non-truthful minus truthful Brier loss on this event gives

$$(0.7 - 0.7)^2 - (0.5 - 0.7)^2 = -0.04.$$

On the event $q_2 = 0.3$, both aggregates equal 0.3. Therefore the non-truthful trace has strictly smaller expected Brier loss. The non-minimum step $G_1 = 1$ is invisible to the scalarized score, so the aggregator can reward inflation of non-minimum prefixes.

B.3 Weighted-average aggregators

Let

$$\Phi_w(F_{1:T}) = \sum_{t=1}^{T} w_t F_t, \qquad w_t \geq 0, \qquad \sum_{t=1}^{T} w_t = 1,$$

with at least two positive weights. Choose indices $i \neq j$ such that $w_i, w_j > 0$, and choose $c \neq 0$ small enough that

$$G_i = q_i + c, \qquad G_j = q_j - \frac{w_i}{w_j} c$$

remain in $(0, 1)$ almost surely, with $G_t = q_t$ for all $t \notin \{i, j\}$. Then

$$\Phi_w(G) = \Phi_w(q) \quad \text{a.s.}$$

while $G \neq q$. Thus any such weighted-average scalarization fails strict propriety for the trace by non-uniqueness.

The strict-suboptimality clause follows under the achievability condition in Theorem B: whenever the truthful weighted aggregate $\Phi_w(q_{1:T})$ differs from the conditionally optimal scalar on a positive-probability event, and an adapted non-truthful trace can make $\Phi_w(G_{1:T})$ equal that optimum, strict propriety of the scalar loss implies

$$\mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_{\Phi_w}(G_{1:T}, Y)] < \mathbb{E}_{\mathbb{P}^\pi}[\mathcal{L}_{\Phi_w}(q_{1:T}, Y)].$$

The average-aggregator construction above is the special case $w_1 = w_2 = \frac{1}{2}$. Therefore the averaging pathology is not specific to the unweighted mean.

No deterministic scalarization used in existing agentic UQ work, including last, average, minimum, or weighted average, strictly elicits the full prefix-conditioned success-probability process. The issue is not that the underlying scalar score is improper; the issue is that scalarization changes the elicited object.

C Censored Proofs

This appendix proves the censored propriety results and records the plug-in approximation criteria used to interpret the exact reduced score.

Proof of Theorem 4.2 (Abstract censored propriety on the observable prefix). For a stopped prefix ending at Z, the observable-prefix complete-data TPS is

$$\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z) = \sum_{t=1}^{Z} w_t S_{\alpha,\beta}(F_t, Y).$$

By complete-data propriety applied to the observable prefix, for any adapted forecast process $F_{1:Z}$,

$$\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(q_{1:Z}, Y; Z)\right] \geq \mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z)\right].$$

By definition of the abstract censored score and the tower property,

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(q_{1:Z}; \mathcal{G}_Z)\right] &= \mathbb{E}_{\mathbb{P}^\pi}\left[\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(q_{1:Z}, Y; Z) \mid \mathcal{G}_Z\right]\right] \\
&= \mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(q_{1:Z}, Y; Z)\right], \\
\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(F_{1:Z}; \mathcal{G}_Z)\right] &= \mathbb{E}_{\mathbb{P}^\pi}\left[\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z) \mid \mathcal{G}_Z\right]\right] \\
&= \mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, Y; Z)\right].
\end{aligned}$$

Substituting these identities into the complete-data inequality gives

$$\mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(q_{1:Z}; \mathcal{G}_Z)\right] \geq \mathbb{E}_{\mathbb{P}^\pi}\left[\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(F_{1:Z}; \mathcal{G}_Z)\right],$$

which proves propriety of the abstract censored TPS on the observable prefix. □

Remark C.1 (Strictness is restricted to the observable prefix). Censored observations contain no information about forecasts after the stopping step. If two forecast processes agree on $F_{1:Z}$ but differ on the unobserved tail $F_{Z+1:T}$, they induce the same conditional projection onto $\mathcal{G}_Z$. Therefore no censored score can be strictly proper for the full unobserved sequence $F_{1:T}$ without additional information about the tail. All strictness statements in the censored setting are consequently restricted to the observable prefix $F_{1:Z}$.

We next record the branch-specific posterior weight that connects the abstract projection to the exact reduced binary form.

Proposition C.2 (Branch-specific posterior identification)**.** *On the censored branch $\Delta = 0$, define*

$$\eta_Z^{(0)} := \mathbb{P}^\pi(Y = 1 \mid \mathcal{G}_Z, \Delta = 0).$$

Under Assumption 2,

$$\eta_Z^{(0)} = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z) = q_Z.$$

Equivalently, the full posterior success weight given $\mathcal{G}_Z$ is

$$\eta_Z^\star := \mathbb{P}^\pi(Y = 1 \mid \mathcal{G}_Z) = \Delta Y + (1 - \Delta) q_Z.$$

Proof. On complete trajectories, $\Delta = 1$, recall that $\mathcal{G}_Z = \sigma(H_Z, Z, \Delta, \Delta Y)$. Thus the stopped-data sigma-field contains the realized outcome through $\Delta Y = Y$. Hence

$$\mathbb{P}^\pi(Y = 1 \mid \mathcal{G}_Z) = Y \qquad \text{on } \{\Delta = 1\}.$$

On censored trajectories, $\Delta = 0$, the outcome is unobserved. By Assumption 2, the administrative stopping information adds no outcome information beyond the stopped history:

$$\mathbb{P}^\pi(Y = 1 \mid \mathcal{G}_Z, \Delta = 0) = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z) = q_Z,$$

where the last equality is the definition of the continuation-success target. Combining the complete and censored branches gives $\eta_Z^\star = \Delta Y + (1 - \Delta) q_Z$. □

Since $Y \in \{0, 1\}$, the abstract censored score admits the binary decomposition

$$\mathrm{TPS}^{\mathrm{cen,abs}}{}_{\alpha,\beta}(F_{1:Z}; \mathcal{G}_Z) = \eta_Z^\star \, \mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, 1; Z) + (1 - \eta_Z^\star) \, \mathrm{TPS}^{\mathrm{obs}}{}_{\alpha,\beta}(F_{1:Z}, 0; Z). \quad (13)$$

Using Proposition C.2, this decomposition is exactly the complete branch plus the q_Z-weighted censored branch in (4).

Proof of Theorem 4.3 (Exact reduced censored beta score). We show that $\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}$ is the explicit reduced form of the abstract conditional projection.

On the complete branch, $\Delta = 1$, the realized outcome Y is observed. Therefore the conditional projection of the observable-prefix complete-data score is simply

$$\sum_{t=1}^{Z} w_t S_{\alpha,\beta}(F_t, Y).$$

On the censored branch, $\Delta = 0$, the outcome is unobserved. By (13) and Proposition C.2, the posterior success weight on the censored branch is $\eta_Z^{(0)} = q_Z$. Thus the conditional projection is

$$\sum_{t=1}^{Z} w_t \left[q_Z S_{\alpha,\beta}(F_t, 1) + (1 - q_Z) S_{\alpha,\beta}(F_t, 0) \right].$$

Combining the complete and censored branches gives exactly

$$\mathrm{TPS}^{\mathrm{cen,exact}}_{\alpha,\beta}(F_{1:Z}; \mathcal{G}_Z) = \sum_{t=1}^{Z} w_t \Big[\Delta S_{\alpha,\beta}(F_t, Y) + (1-\Delta)\big(q_Z S_{\alpha,\beta}(F_t, 1) + (1-q_Z) S_{\alpha,\beta}(F_t, 0) \big) \Big],$$

which is (4).

At the logarithmic endpoint, the exact reduced score is

$$\mathrm{TPS}_{\log}^{\mathrm{cen,exact}}(F_{1:Z};\mathcal{G}_Z)=\sum_{t=1}^{Z} w_t\Big[\Delta\{Y\log F_t+(1-Y)\log(1-F_t)\} \\ +(1-\Delta)\{q_Z\log F_t+(1-q_Z)\log(1-F_t)\}\Big]. \tag{14}$$

Propriety follows because this score is algebraically the abstract censored projection from Theorem 4.2. Hence its expected score is maximized by the truthful process $q_{1:Z}$.

If $S_{\alpha,\beta}$ is strictly proper and $w_t>0$ for all observed steps, strictness holds on the observable prefix. By the tower-property identities above, the expected regret of $\mathrm{TPS}_{\alpha,\beta}^{\mathrm{cen,exact}}$ equals the expected observable-prefix complete-data regret. Although Y is unobserved on censored branches, it is well defined under the full-data law. Thus strictness can be checked through the complete-data regret.

For each step, define the conditional proper regret

$$R_t(F_t):=q_t\{S_{\alpha,\beta}(q_t,1)-S_{\alpha,\beta}(F_t,1)\}+(1-q_t)\{S_{\alpha,\beta}(q_t,0)-S_{\alpha,\beta}(F_t,0)\}.$$

By strict propriety of $S_{\alpha,\beta}$, $R_t(F_t)\geq 0$, with equality if and only if $F_t=q_t$ almost surely. The expected observable-prefix complete-data regret is

$$\mathbb{E}_{\mathbb{P}^\pi}\left[\sum_{t=1}^{T}\mathbf{1}\{t\leq Z\}w_tR_t(F_t)\right].$$

Every term in this sum is nonnegative, and $w_t>0$ on observed steps. If the total regret is zero, then $R_t(F_t)=0$ almost surely on $\{t\leq Z\}$. Therefore

$$F_t=q_t \qquad \text{a.s. on } \{t\leq Z\}.$$

Thus $\mathrm{TPS}_{\alpha,\beta}^{\mathrm{cen,exact}}$ is strictly proper for the observable prefix, but not for the unobserved tail. □

Proposition C.3 (Target of the simple censored score)**.** *For any strictly proper binary score $S_{\alpha,\beta}$, the simple censored score in* (5) *is strictly proper for the observable pseudo-label target*

$$m_t:=\mathbb{E}_{\mathbb{P}^\pi}[\Delta Y\mid\mathcal{H}_t]=\mathbb{P}^\pi(\Delta=1,Y=1\mid\mathcal{H}_t),$$

on the observed prefix. It is strictly proper for the original continuation-success target $q_t=\mathbb{P}^\pi(Y=1\mid\mathcal{H}_t)$ if and only if

$$\mathbb{P}^\pi(\Delta=0,Y=1\mid\mathcal{H}_t)=0$$

almost surely for the observed steps; equivalently, whenever $\mathbb{P}^\pi(Y=1\mid\mathcal{H}_t)>0$, $\mathbb{P}^\pi(\Delta=0\mid\mathcal{H}_t,Y=1)=0$.

Proof of Proposition C.3. The simple censored score replaces the unobserved outcome by the pseudo-label

$$\widetilde{Y}^{\mathrm{simple}}:=\Delta Y.$$

For the logarithmic endpoint, the per-step simple score is

$$S_t^{\mathrm{simple}}=\Delta Y\log F_t+(1-\Delta Y)\log(1-F_t).$$

Conditioning on the available history gives

$$\mathbb{E}_{\mathbb{P}^\pi}[S_t^{\mathrm{simple}}\mid\mathcal{H}_t]=m_t\log F_t+(1-m_t)\log(1-F_t),\qquad m_t:=\mathbb{E}_{\mathbb{P}^\pi}[\Delta Y\mid\mathcal{H}_t].$$

This conditional expected log score is uniquely maximized at

$$F_t=m_t.$$

The same argument applies to any strictly proper binary score $S_{\alpha,\beta}$. Conditional on $\mathcal{H}_t$, the pseudo-label ΔY has Bernoulli mean m_t. A strictly proper binary score is therefore uniquely optimized by reporting m_t. Summing over observed steps with positive weights preserves strict propriety for the observable pseudo-label target m_t.

Finally,

$$q_t - m_t = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_t) - \mathbb{P}^\pi(\Delta = 1, Y = 1 \mid \mathcal{H}_t) = \mathbb{P}^\pi(\Delta = 0, Y = 1 \mid \mathcal{H}_t).$$

Thus the simple censored score elicits the original continuation-success target q_t exactly if and only if

$$\mathbb{P}^\pi(\Delta = 0, Y = 1 \mid \mathcal{H}_t) = 0$$

almost surely, equivalently

$$\mathbb{P}^\pi(\Delta = 0 \mid \mathcal{H}_t, Y = 1) = 0$$

on histories where $\mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_t) > 0$. This is precisely the zero missing-success-mass regime represented by the failure-side $q_Z \approx 0$ approximation on censored prefixes. □

Remark C.4 (Plug-in propriety criterion). Suppose the exact censored score is implemented with an estimated continuation-success weight $\widehat{q}_Z$ on censored prefixes. Define the soft surrogate outcome

$$\widetilde{Y}_Z := \Delta Y + (1 - \Delta)\widehat{q}_Z.$$

At an observed step $t \le Z$, the plug-in per-step score has conditional mean

$$\mathbb{E}_{\mathbb{P}^\pi}\left[\widetilde{Y}_Z S_{\alpha,\beta}(p, 1) + (1 - \widetilde{Y}_Z)S_{\alpha,\beta}(p, 0) \mid \mathcal{H}_t\right].$$

Let

$$a_t := \mathbb{E}_{\mathbb{P}^\pi}[\widetilde{Y}_Z \mid \mathcal{H}_t].$$

By binary strict propriety, the conditional expected plug-in score is optimized at $p = a_t$. Therefore the plug-in score preserves propriety for the original target q_t at step t if and only if

$$\mathbb{E}_{\mathbb{P}^\pi}[\widetilde{Y}_Z \mid \mathcal{H}_t] = q_t.$$

In particular, if $\widehat{q}_Z = q_Z$ on censored prefixes, then

$$\widetilde{Y}_Z = \Delta Y + (1 - \Delta)q_Z = \mathbb{E}_{\mathbb{P}^\pi}[Y \mid \mathcal{G}_Z].$$

Since $\mathcal{H}_t \subseteq \mathcal{G}_Z$ on observed steps $t \le Z$, the tower property gives

$$\mathbb{E}_{\mathbb{P}^\pi}[\widetilde{Y}_Z \mid \mathcal{H}_t] = \mathbb{E}_{\mathbb{P}^\pi}[\mathbb{E}_{\mathbb{P}^\pi}[Y \mid \mathcal{G}_Z] \mid \mathcal{H}_t] = \mathbb{E}_{\mathbb{P}^\pi}[Y \mid \mathcal{H}_t] = q_t.$$

Thus exact q_Z estimation is a clean sufficient condition for preserved propriety, while the general criterion is the conditional-mean equality above.

Remark C.5 (Regret decomposition under plug-in error). Let

$$r_t := \mathbb{E}_{\mathbb{P}^\pi}[\widetilde{Y}_Z \mid \mathcal{H}_t] - q_t$$

be the conditional-mean bias of the plug-in surrogate at an observed step. For any $p \in [0, 1]$, define the binary score contrast

$$D(p) := S_{\alpha,\beta}(p, 1) - S_{\alpha,\beta}(p, 0).$$

The per-step expected plug-in regret relative to reporting p decomposes as

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}^\pi}\left[\widetilde{Y}_Z\{S_{\alpha,\beta}(q_t, 1) - S_{\alpha,\beta}(p, 1)\} + (1 - \widetilde{Y}_Z)\{S_{\alpha,\beta}(q_t, 0) - S_{\alpha,\beta}(p, 0)\} \mid \mathcal{H}_t\right] \\ &\quad = R_t^\star(p) + r_t\big(D(q_t) - D(p)\big),\end{aligned}$$

where

$$R_t^\star(p) := q_t\{S_{\alpha,\beta}(q_t, 1) - S_{\alpha,\beta}(p, 1)\} + (1 - q_t)\{S_{\alpha,\beta}(q_t, 0) - S_{\alpha,\beta}(p, 0)\} \ge 0$$

is the oracle proper regret. For interior bounded beta-family members ($\alpha, \beta > 0$), the contrast D is bounded by some constant $B_{\alpha,\beta} < \infty$, so

$$\big|r_t\big(D(q_t) - D(p)\big)\big| \le 2B_{\alpha,\beta}|r_t|.$$

After summing with weights w_t, this gives an approximate-propriety bound over the observable prefix: the plug-in score differs from the oracle proper regret by a perturbation proportional to the conditional-mean bias $|r_t|$. The logarithmic endpoint is unbounded, so this bounded-error statement applies only to interior beta-family members.

For the log score there is a simpler, exact statement for each censored prefix. Write $L_z(q)$ for the exact censored log score of a prefix when its continuation probability is q. For $0 < F_t < 1$, this score is a straight line in q:

$$L_z(\widehat{q}_z) - L_z(q_z) = (\widehat{q}_z - q_z)\, c_z, \qquad c_z = \sum_{t=1}^{z} w_t \log \frac{F_t}{1 - F_t}. \tag{15}$$

An error in the estimate of q_z therefore shifts the score by that error times the slope c_z, the weighted sum of the prefix's log-odds. The slope depends only on the observed forecasts and weights, so it can be computed without rerunning anything. Two consequences follow. First, the score stays within ε of its exact value whenever $|\widehat{q}_z - q_z| \le \varepsilon/|c_z|$; if $c_z = 0$, the score does not depend on q_z at all. Second, because the score is linear in q, its values at $q = 0$ and $q = 1$ bound the exact score for every $q \in [0, 1]$. An estimate of q_z that is correct on average also gives the correct score on average. These statements hold for each prefix separately; they are not a uniform bound for the unbounded log score.

D Calibration Procedure

We recalibrate per-step forecast streams using single-split cross-fitted Platt scaling. The procedure is applied consistently across the experiments whenever a Platt-calibrated predictor variant is reported. This appendix documents the split construction, logit-space feature transform, weighted logistic fit, monotonicity safeguard, and cross-fitting procedure.

Cross-fitting design. We use a single $50/50$ trajectory-level split rather than K-fold cross-fitting so that each calibrated stream is produced by only two held-out calibration maps. With K independent fits on weak signals, fold-specific slopes can be unstable or sign-inconsistent; concatenating those maps can introduce non-monotone calibrated outputs.

Splitting. Trajectories are partitioned into two equally sized halves stratified by outcome label Y. Within each label, trajectories are ordered by a stable trajectory identifier and alternated between halves, yielding approximately balanced success/failure distributions in both splits. All per-step records inherit the split assignment of their parent trajectory, so no within-trajectory information leaks across the calibration split.

Feature construction. Each raw per-step forecast F_t is first clipped into $[\varepsilon, 1 - \varepsilon]$ with $\varepsilon = 10^{-6}$ and mapped to log-odds:

$$x_{\text{raw}} = \text{logit}(F_t) = \log\left(\frac{F_t}{1 - F_t}\right).$$

Within each training split, a weighted mean μ_{train} and weighted standard deviation σ_{train} are computed from x_{raw} using the same linear-front step weights w_t used by TPS. The standardized feature is

$$x = \frac{x_{\text{raw}} - \mu_{\text{train}}}{\max(\sigma_{\text{train}}, 10^{-6})}.$$

Held-out trajectories use the training split's μ_{train} and σ_{train}; no statistics from the held-out half enter the standardization.

Weighted logistic fit. Within each training split we fit

$$p_{\text{cal}} = \text{sigmoid}(a + b\, x)$$

by weighted maximum likelihood on per-step records, where $\text{sigmoid}(u) = (1 + e^{-u})^{-1}$. The fit uses sample weights w_t and an L_2 penalty of $\lambda = 1.0$ on the slope b only. Sample weights are not class-balanced; the calibrator targets the empirical Bernoulli success labels rather than an artificial $50/50$ prior.

Monotone non-decreasing fallback. If the fitted slope satisfies $b < 0$, we set $b = 0$ and

$$a = \text{logit}(\bar{y}_{\text{train}}),$$

where $\bar{y}_{\text{train}}$ is the weighted in-train success rate. This enforces a non-decreasing mapping and prevents pathological sign flips on signals with no usable resolution. The intercept-only fallback returns the in-train base rate at every input. After computing p_{cal}, a final ε-clip to $[10^{-6}, 1 - 10^{-6}]$ is applied before scoring.

Cross-fitting. The model fitted on split A is applied to split B, and the model fitted on split B is applied to split A. Each trajectory is therefore calibrated by a model that did not observe its parent split during training.

Numerical stabilization. Verbal-confidence streams concentrate near $\{0.90, 0.95, 1.00\}$, so their log-odds span a range dominated by the ε-clip jump from $\mathrm{logit}(0.95) \approx 2.94$ to $\mathrm{logit}(1-\varepsilon) \approx 13.82$. Per-split standardization prevents this clipped boundary from dominating the logistic fit. The L_2 penalty $\lambda = 1.0$ is used to shrink unstable slopes on small calibration splits with weakly informative signals. Calibrating with the same linear-front weights as TPS aligns the calibrator with the weighted objective used during evaluation.

Diagnostics on Tau2-Bench verbal confidences. The procedure was applied to the $n = 201$ uncensored Tau2-Bench trajectories used in the calibration-invariance experiment, with empirical success rate $\bar{y} = 0.443$.

Quantity	Value
Split A slope b	0.2639 (no fallback)
Split B slope b	0.2213 (no fallback)
Fallback triggered	No
Raw values $0.90, 0.95, 1.00$ map to	$0.33, 0.34, 0.47$
Calibrated probability range	$[0.33, 0.47]$

Table 1: Single-split Platt calibration diagnostics on Tau2-Bench verbal confidences. Both splits return positive finite slopes, so the monotone fallback is not triggered.

The corresponding metric changes are summarized in Section 6.1; this table reports only calibration diagnostics.

D.1 Alternative-calibrator sensitivity

To check that the result in Section 6.1 is not specific to Platt scaling, we recalibrate the same Tau2 verbal-confidence stream with two other common methods, isotonic regression and beta calibration, using the same folds and weights. Table 2 shows that all three give essentially the same 5.67-nat improvement in log TPS, while the change in AUROC is small and its 95% interval includes zero in every case.

Calibrator	$\Delta\mathrm{TPS}_{\log}$ [95% CI]	$\Delta\mathrm{AUROC}$ [95% CI]
Platt	$+5.6717$ $[5.3944, 5.9063]$	-0.0089 $[-0.0292, +0.0124]$
Isotonic	$+5.6696$ $[5.3940, 5.9048]$	-0.0097 $[-0.0276, +0.0086]$
Beta	$+5.6701$ $[5.3942, 5.9050]$	-0.0072 $[-0.0253, +0.0115]$

Table 2: Alternative calibrators on the 201 Tau2 tasks. Each entry is the change from the raw to the calibrated stream. Intervals use 1,000 paired task-level bootstrap resamples with the calibration folds and fitted maps held fixed, so they do not include the uncertainty from refitting the calibrator.

Limitations. Platt scaling is a simple calibrator, not an optimal one. The comparison above covers three calibrators on one stream, Tau2 verbal confidence, rather than every predictor and dataset. The fixed-rank analysis in Appendix E removes the calibrator from the argument entirely by holding the rank-metric input fixed.

The procedure is validated primarily on the Tau2 verbal-confidence stream. Other predictors admit the same procedure mechanically, but their slopes and calibrated ranges differ and are reported in Appendix F. Weighted standard deviation can be small on highly saturated streams; in that case the $\sigma_{\mathrm{train}} \geq 10^{-6}$ floor activates and standardization degenerates. This floor was not active in our runs, but it is a known failure mode for streams with effectively a single unique value. When the evaluator weight schedule changes, for example from linear-front to uniform, the calibrator must be refit with matching weights; otherwise the calibration-target mismatch is conflated with the intended score-family or weighting comparison. Finally, the cross-fitted procedure produces two distinct (a, b) pairs for the same predictor, so trajectories in different splits are calibrated under slightly different

mappings. This is a known property of held-out calibration rather than an error: every reported calibrated trajectory is scored under a map fitted without using that trajectory's outcome.

E Fixed-Rank Sensitivity Analysis

To check that the calibration-invariance gap in Section 6.1 is not a Platt-scaling artifact, we repeat the comparison with the input to the rank-based metrics held fixed. On the same Tau2 traces, we compute one trajectory scalar $r_i = \Phi(F^{\mathrm{raw}}_{i,1:T_i})$ from the raw stream and reuse this same r_i for AUROC, AUPRC, and AURC in every row. Only the per-step probability stream evaluated by $\mathrm{TPS}_{\log}$ is transformed.

We apply five transforms chosen to span qualitatively distinct probability-scale distortions: identity, affine compression $g(x) = 0.4 + 0.2x$, square root, square, and the empirical Platt mapping from Section 6.1, covering linear, sublinear, superlinear, and learned rescalings. AUROC, AUPRC, and AURC are identical by construction across all five, while $\mathrm{TPS}_{\log}$ spans 5.7 nats. The label-free affine compression provides the cleanest contrast: it shifts the probability scale without using any outcome information, yet the rank metrics remain fixed. The gap is structural to rank-based evaluation, not a property of any particular calibrator.

F Predictor Transparency Tables

This appendix reports complete-observation predictor transparency tables for StrategyQA, Tau2-Bench, and HotpotQA. All rows in this appendix have an observed terminal outcome Y; administratively censored and informatively terminated trajectories are excluded here and handled separately in Appendices I.1, I.2, and G. Thus, this appendix is restricted to the uncensored evaluation setting of §4.1.

These tables are provided for transparency rather than for predictor selection. They document how the reporting convention of §5.2 applies across the full predictor pool and across three complete-observation datasets. They also distinguish per-step probability streams, which are valid inputs to TPS, from AUQ-style scalar aggregators, which collapse a trajectory before evaluation and therefore fall outside the domain of TPS.

Sample definition. The transparency tables are restricted to completed trajectories for which the benchmark returned a terminal success/failure label. The resulting samples are StrategyQA ($n = 2229$), Tau2-Bench ($n = 201$), and HotpotQA ($n = 1529$). Trajectories ending because of step-limit exhaustion, parser failure, tool-interface failure, or environment termination without a clean task label are excluded from this appendix and handled in the termination audit and censored-analysis appendices. For predictors defined only on action spans, the available sample may be slightly smaller; such cases are noted in the corresponding table captions.

Per-step predictor pool. For each dataset, we report five per-step confidence streams, all stored on $[0, 1]$ with larger values indicating higher predicted probability of eventual task success. These are: verbal confidence; completion-token probability; completion entropy confidence; action-span token probability; and action-span entropy confidence. The completion-level streams are computed over the full generated step, while the action-span streams restrict the signal to the parsed action that the agent commits to executing. We also include a constant empirical base-rate stream, $F_t \equiv \bar{y}$, as an uninformative reference. Each non-constant stream is shown in raw form and, when available, after single-split cross-fitted Platt recalibration. The recalibration procedure is described in Appendix D.

Verbal confidence is the scalar probability of success that the model self-reports at each step, parsed from the structured confidence field and clipped to $[0, 1]$; missing or malformed confidence fields yield a missing per-step value. *Completion-token probability* is the mean chosen-token probability across all tokens generated at that step, capturing how peaked the next-token distribution was on average over the model's entire reasoning-plus-action output. *Completion-entropy confidence* measures the same step's distributional uncertainty as the average per-token Shannon entropy over the top-5 candidates, in nats, and inverts it to a confidence via $1 - H/\ln 5$. *Action-span token probability* and *action-span entropy confidence* are the same two signals restricted to the tokens in the parsed committed action: the completion variants reflect confidence over the entire deliberation, while the action-span variants isolate confidence in the action the agent actually executes.

Metrics. All metric columns carry an explicit orientation arrow. AUROC and AUPRC are failure-oriented, with positive class $Y = 0$, and are computed from a scalar trajectory risk summary. AURC is reported as a risk–coverage diagnostic, with lower values better. Trajectory ECE is reported as tie-aware quantile-binned calibration error with 10 bins. Scalarized trajectory Brier is the AUQ-style trajectory reliability diagnostic computed on the same scalar trajectory summary. The TPS columns are reward-oriented, so higher values are better, and score the full per-step probability stream using normalized linear-front weights.

A small AUROC change between raw and Platt rows should not be read as a violation of AUROC's monotone-invariance property. That invariance applies to a single global monotone transformation. Our Platt rows are cross-fitted: separate fold-specific calibration maps are fit and then pooled, so inter-fold scale differences can perturb pooled trajectory rankings slightly. The fixed-rank robustness check in Appendix E isolates the theoretical invariance case by holding the AUROC/AUPRC/AURC input scalar fixed by construction.

Scalar AUQ-style aggregators. AUQ-style scalar aggregators such as Φ_{last}, Φ_{avg}, and Φ_{min} produce one scalar per trajectory rather than an adapted per-step probability stream. They are therefore not scored by TPS. We report them in a separate scalar-predictor table using only scalar-compatible diagnostics: AUROC, AUPRC, AURC, trajectory ECE, and scalarized trajectory Brier. The TPS entries are omitted because these scalar predictors do not claim to provide the full prefix-conditioned process $(F_t)_{t=1}^{T}$.

F.1 Per-step predictor transparency

Tables 3–5 report complete-observation metrics for per-step confidence streams. These predictors fall within the domain of TPS, because they provide a probability-like value at each trajectory prefix. For all three tables, AUROC, AUPRC, and AURC are failure-oriented scalar diagnostics; T-ECE is tie-aware quantile-binned calibration error with 10 bins; T-Brier is a scalar trajectory diagnostic; and the TPS columns score the full per-step stream using normalized linear-front weights. The scalar diagnostics summarize each trajectory by the linear-front weighted root-mean-square of its forecasts. The one exception is the two Tau2 verbal-confidence AUROC entries, which use the weighted mean so that they match Appendix D.1.

Predictor	AUROC ↑	AUPRC ↑	AURC ↓	T-ECE ↓	T-Brier ↓	$\mathrm{TPS}_{\log}$ ↑	$\mathrm{TPS}_{\mathrm{Brier}}$ ↑	$\mathrm{TPS}_{\beta(2,4)}$ ↑
Verbal confidence (raw)	0.701	0.354	0.092	0.201	0.170	-0.569	-0.178	-0.00387
Verbal confidence (Platt)	0.709	0.334	0.090	0.055	0.128	-0.425	-0.128	-0.00262
Completion entropy confidence (raw)	0.598	0.199	0.115	0.022	0.131	-0.434	-0.132	-0.00263
Completion entropy confidence (Platt)	0.599	0.197	0.115	0.029	0.131	-0.432	-0.132	-0.00263
Completion token probability (raw)	0.594	0.193	0.116	0.060	0.135	-0.453	-0.136	-0.00263
Completion token probability (Platt)	0.595	0.192	0.116	0.030	0.132	-0.433	-0.132	-0.00263
Action-span entropy confidence (raw)	0.537	0.173	0.144	0.081	0.140	-0.817	-0.145	-0.00263
Action-span entropy confidence (Platt)	0.556	0.187	0.140	0.022	0.133	-0.436	-0.133	-0.00263
Action-span token probability (raw)	0.543	0.176	0.141	0.106	0.144	-0.921	-0.148	-0.00263
Action-span token probability (Platt)	0.553	0.185	0.139	0.019	0.133	-0.436	-0.133	-0.00263
Base-rate constant	0.500	0.158	0.158	0.000	0.133	-0.436	-0.133	-0.00263

Table 3: StrategyQA complete-observation predictor transparency ($n = 2229$, $\bar{y} = 0.842$). The table reports the per-step confidence streams evaluated under the conventions stated above.

F.2 Scalar AUQ-style predictors

Table 6 reports scalar AUQ-style trajectory predictors. These rows collapse a trajectory to a single scalar before evaluation, so they are reported only on scalar-compatible diagnostics: AUROC, AUPRC, AURC, T-ECE, and T-Brier. T-ECE again denotes tie-aware quantile-binned calibration error with 10 bins. These predictors are outside the domain of TPS, which scores adapted per-step probability streams.

G Termination Mechanisms and Parse-Error Sensitivity

The censored trajectory extension requires that censored trajectories are stopped by an *administrative* mechanism: the stopping event is externally imposed and is not itself evidence about the unobserved

Predictor	AUROC ↑	AUPRC ↑	AURC ↓	T-ECE ↓	T-Brier ↓	$\mathrm{TPS}_{\log}$ ↑	$\mathrm{TPS}_{\mathrm{Brier}}$ ↑	$\mathrm{TPS}_{\beta(2,4)}$ ↑
Verbal confidence (raw)	0.616	0.668	0.479	0.537	0.532	-6.351	-0.543	-0.00929
Verbal confidence (Platt)	0.608	0.693	0.490	0.082	0.240	-0.679	-0.243	-0.00744
Completion entropy confidence (raw)	0.558	0.620	0.509	0.463	0.460	-1.419	-0.465	-0.00927
Completion entropy confidence (Platt)	0.552	0.591	0.526	0.066	0.246	-0.687	-0.247	-0.00760
Completion token probability (raw)	0.543	0.613	0.520	0.495	0.491	-1.641	-0.494	-0.00928
Completion token probability (Platt)	0.534	0.579	0.536	0.045	0.247	-0.687	-0.247	-0.00760
Action-span entropy confidence (raw)	0.583	0.643	0.483	0.503	0.496	-4.083	-0.518	-0.00928
Action-span entropy confidence (Platt)	0.583	0.658	0.518	0.082	0.245	-0.685	-0.246	-0.00757
Action-span token probability (raw)	0.567	0.633	0.491	0.521	0.516	-4.712	-0.531	-0.00929
Action-span token probability (Platt)	0.587	0.666	0.513	0.105	0.245	-0.685	-0.246	-0.00756
Base-rate constant	0.500	0.557	0.557	0.000	0.247	-0.687	-0.247	-0.00760

Table 4: Tau2-Bench complete-observation predictor transparency ($n = 201$, $\bar{y} = 0.443$). Recalibrating verbal confidence changes $\mathrm{TPS}_{\log}$ sharply but leaves the rank-based diagnostics nearly unchanged. AUROC moves slightly ($0.616 \to 0.608$; $\Delta = -0.0089$, Appendix D.1) because cross-fitting uses a separate calibration map for each half of the data, which can reorder trajectories across halves; a single monotone transformation would leave AUROC unchanged. The two verbal-confidence AUROC entries use the weighted-mean trajectory summary of Appendix D.1; all other scalar diagnostics use the weighted-RMS summary.

Predictor	AUROC ↑	AUPRC ↑	AURC ↓	T-ECE ↓	T-Brier ↓	$\mathrm{TPS}_{\log}$ ↑	$\mathrm{TPS}_{\mathrm{Brier}}$ ↑	$\mathrm{TPS}_{\beta(2,4)}$ ↑
Verbal confidence (raw)	0.556	0.463	0.377	0.318	0.344	-2.642	-0.356	-0.00689
Verbal confidence (Platt)	0.544	0.477	0.385	0.075	0.242	-0.677	-0.242	-0.00647
Completion entropy confidence (raw)	0.596	0.504	0.346	0.321	0.343	-1.048	-0.345	-0.00693
Completion entropy confidence (Platt)	0.596	0.507	0.346	0.037	0.237	-0.671	-0.239	-0.00640
Completion token probability (raw)	0.588	0.494	0.348	0.352	0.365	-1.201	-0.367	-0.00694
Completion token probability (Platt)	0.589	0.499	0.348	0.036	0.238	-0.673	-0.240	-0.00642
Action-span entropy confidence (raw)	0.541	0.440	0.374	0.371	0.380	-2.135	-0.388	-0.00694
Action-span entropy confidence (Platt)	0.585	0.486	0.351	0.043	0.240	-0.675	-0.241	-0.00646
Action-span token probability (raw)	0.538	0.438	0.375	0.386	0.392	-2.517	-0.398	-0.00695
Action-span token probability (Platt)	0.575	0.476	0.356	0.032	0.240	-0.676	-0.241	-0.00646
Base-rate constant	0.500	0.417	0.417	0.000	0.243	-0.679	-0.243	-0.00649

Table 5: HotpotQA complete-observation predictor transparency ($n = 1529$, $\bar{y} = 0.583$). The small AUROC shift for verbal confidence ($0.556 \to 0.544$) is due to cross-fitted Platt calibration pooling fold-specific monotone maps, which can perturb inter-fold rankings; this does not contradict the single global monotone-transform invariance used in the calibration-invariance argument. Action-span rows use the available complete action-span subset where applicable.

final outcome beyond what is already contained in the observed prefix. In our benchmark setting, step-budget terminations satisfy this assumption most plausibly because the stopping rule is fixed by the evaluation harness in advance, independently of the agent's behaviour.

Parser failures, by contrast, are treated as informative failures of the interaction protocol rather than administrative censoring. A parser failure occurs because the agent emitted malformed or non-executable output. This is not merely an unobserved continuation; it is evidence that the agent was struggling with the task or tool interface. Including such trajectories as right-censored observations would violate Assumption 2 by mixing model-formatting failures with task-level uncertainty calibration. We therefore exclude parser failures from theorem-backed censored scoring and report their frequency here as an assumption-discipline audit.

We separate terminations into three categories:

- **Benchmark-completed trajectories.** The benchmark produced a proper terminal outcome $Y \in \{0, 1\}$. These trajectories enter complete-observation scoring ($\delta = 1$).
- **Step-budget trajectories.** The trajectory reached the fixed benchmark step budget before the benchmark produced a terminal outcome. The stopping rule is externally imposed and budget-determined, so these trajectories are eligible for censored scoring ($\delta = 0$, administratively censored).
- **Parser-failure trajectories.** The agent emitted malformed output that the harness could not execute or score. The stopping mechanism is informative, so these trajectories are excluded from both complete-observation and theorem-backed censored scoring.

Dataset	Scalar predictor	AUROC ↑	AUPRC ↑	AURC ↓	T-ECE ↓	T-Brier ↓
StrategyQA	AUQ Φ_{last} raw	0.752	0.349	0.083	0.089	0.128
StrategyQA	AUQ Φ_{avg} raw	0.717	0.367	0.088	0.126	0.138
StrategyQA	AUQ Φ_{min} raw	0.684	0.300	0.101	0.306	0.235
StrategyQA	AUQ Φ_{last} Platt	0.749	0.366	0.081	0.061	0.120
StrategyQA	AUQ Φ_{avg} Platt	0.714	0.335	0.089	0.044	0.123
StrategyQA	AUQ Φ_{min} Platt	0.682	0.288	0.099	0.036	0.126
Tau2-Bench	AUQ Φ_{last} raw	0.536	0.589	0.539	0.555	0.552
Tau2-Bench	AUQ Φ_{avg} raw	0.628	0.704	0.478	0.549	0.545
Tau2-Bench	AUQ Φ_{min} raw	0.606	0.628	0.500	0.511	0.502
Tau2-Bench	AUQ Φ_{last} Platt	0.534	0.596	0.540	0.001	0.239
Tau2-Bench	AUQ Φ_{avg} Platt	0.604	0.682	0.494	0.079	0.240
Tau2-Bench	AUQ Φ_{min} Platt	0.572	0.618	0.512	0.060	0.243
HotpotQA	AUQ Φ_{last} raw	0.546	0.458	0.394	0.409	0.407
HotpotQA	AUQ Φ_{avg} raw	0.564	0.475	0.374	0.345	0.360
HotpotQA	AUQ Φ_{min} raw	0.557	0.456	0.383	0.280	0.326
HotpotQA	AUQ Φ_{last} Platt	0.546	0.459	0.394	0.104	0.256
HotpotQA	AUQ Φ_{avg} Platt	0.545	0.468	0.384	0.034	0.241
HotpotQA	AUQ Φ_{min} Platt	0.546	0.460	0.387	0.033	0.241

Table 6: Scalar AUQ-style trajectory predictors on the complete-observation samples. These predictors are not assigned TPS values because they collapse the trajectory before evaluation and therefore do not provide the full prefix-conditioned probability stream $(F_t)_{t=1}^{T}$.

Step budgets. The maximum step budgets were fixed in advance for each benchmark: 7 steps for HotpotQA, 16 steps for StrategyQA, 50 steps for Tau2-Bench, and 30 steps for WebShop. These budgets were empirically chosen based on benchmark difficulty and then held fixed across trajectories within each benchmark. Thus, a step-budget termination means that a trajectory reached the benchmark-specific fixed budget before a terminal success/failure label was produced; it does not mean that the evaluator stopped the trajectory adaptively based on the agent's apparent performance.

G.1 Termination audit

Table 7 reports the stop-reason breakdown for every dataset used in this paper. The `max_steps` column counts trajectories that reached the benchmark-specific step budget stated above.

Dataset	Step budget	Total	Clean Y	`max_steps`	`parse_error`	Other	Uncensored n
StrategyQA	16	2,290	2,229	59	2	0	2,229
Tau2-Bench	50	278	201	1	0	76	201
HotpotQA	7	2,000	1,529	458	13	0	1,529
WebShop	30	500	163	145	192	0	163

Table 7: Termination audit by benchmark. Step budgets are fixed ex ante for each dataset. WebShop is the only dataset used for the natural-censoring analysis.

For Tau2-Bench, the 76 "Other" non-clean terminations consist of 70 `tool_error` and 6 `env_terminated` trajectories. Because no final task label is observed and the stop reason is not a fixed administrative budget, these trajectories are excluded from both the uncensored transparency sample and theorem-backed censored scoring.

The predictor-transparency appendix uses only trajectories with clean terminal outcomes for StrategyQA, Tau2-Bench, and HotpotQA. Thus, the sample sizes in Appendix F correspond to the rows with observed Y: StrategyQA $n = 2229$, Tau2-Bench $n = 201$, and HotpotQA $n = 1529$, with

minor predictor-level variation when action-span signals are unavailable for a row. The WebShop natural-censoring experiment uses the working sample of 308 trajectories after excluding parser failures: 163 complete trajectories with observed Y, plus 145 step-budget trajectories treated as administratively censored.

G.2 Parse-error sensitivity analysis

The consequential parse-error case is WebShop. Of 500 attempted WebShop trajectories, 192 ended in parser failure, 163 terminated with an observed outcome, and 145 hit the fixed step budget. The censored WebShop analysis therefore uses the working sample

$$n_{\text{work}} = 163 + 145 = 308,$$

with administrative-censoring rate

$$\frac{145}{308} = 47.08\%,$$

and excludes the 192 parser-failure trajectories.

This exclusion follows directly from Assumption 2. Step-budget termination is imposed by the benchmark and is therefore a plausible administrative censoring mechanism. Parser failure, however, is generated by the agent's own malformed output and is plausibly correlated with task difficulty, tool-use failure, and eventual success probability. Treating parser failures as right-censored observations would convert an informative failure mode into an administrative censoring event.

As a contra-assumption sensitivity check, we re-ran the primary WebShop configuration (verbal confidence, log score, linear-front weights) treating all 192 WebShop parser-failure trajectories as administratively censored at the step where parsing failed. This inflates $\widehat{\text{TPS}}$ by approximately $0.04\,\text{nats}$ relative to the working-sample estimate, consistent with the directional prediction: parser-failure trajectories tend to be long and low-confidence, so including them shifts the sample mean in the same direction as the assumption-consistent censored subset. The qualitative conclusion is unchanged, but the result confirms that the assumption-disciplined working sample is the conservative analysis choice.

H q_Z Estimation and HotpotQA Audit

The exact reduced censored score in Theorem 4.3 depends on the stopped-prefix continuation-success probability

$$q_Z = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z).$$

This appendix describes operational estimators for q_Z and reports a Monte Carlo continuation audit on naturally max-step HotpotQA prefixes.

H.1 Operational strategies for $\widehat{q}_Z$

There are three practical ways to estimate the nuisance quantity q_Z.

Direct Monte Carlo continuation. Resume each censored prefix under the same model, harness, and stopping rule, and estimate

$$\widehat{q}_Z^{\text{MC}} = \frac{1}{M} \sum_{m=1}^{M} Y^{(m)}.$$

This is the direct rollout estimator of the conditional terminal-success probability from the stopped prefix. It is the analogue of Monte Carlo policy evaluation, where expected returns are estimated by averaging sampled returns, and it is also used in recent LLM-agent work to estimate step-level rewards by continuation sampling [4, 26]. It is best suited to benchmarks where the prefix state can be resumed exactly and continuation rollouts are inexpensive.

Plug-in nuisance model. Train a separate estimator

$$\widehat{q}_Z^{\text{plug}} = g_\psi(H_Z, Z, \text{metadata})$$

on prefixes with known continuation outcomes, then apply it to censored prefixes. This treats $q_Z = \mathbb{P}^\pi(Y = 1 \mid \mathcal{H}_Z)$ as a conditional-mean nuisance function. Estimating such nuisance functions

separately from the target score is standard in semiparametric and missing-data settings, and modern practice often uses flexible ML nuisance estimators with sample splitting or cross-fitting to reduce overfitting bias [27, 28]. In our setting, the nuisance model must be kept separate from the uncertainty predictor being evaluated; otherwise the evaluator and predictor under test are no longer cleanly separated. Remark C.4 gives the conditional-mean criterion required for plug-in exactness.

Hybrid audit. Run Monte Carlo continuation on a subset of censored prefixes, train a plug-in nuisance model on those audited labels, and apply the plug-in estimator to the full censored set. This combines the rollout-based estimate of q_Z with a scalable nuisance-modeling step, following the same principle as using audited outcomes to train an outcome-regression nuisance model [27, 28].

Which option to use depends on what the deployment allows. If stopped prefixes can be resumed reliably, rerun them and report the exact score; the audit below shows that a handful of reruns per prefix is enough for conclusions about a whole dataset, although judging an individual prefix needs more. If prefixes cannot be resumed, a trained estimate of q_Z can be used, provided it is kept separate from the predictor being evaluated and satisfies the condition in Remark C.4. If neither is possible, report the scores at $q_Z = 0$ and $q_Z = 1$ as a range, and label any single $\mathrm{TPS}^{\mathrm{cen,simple}}$ value as the $q_Z \approx 0$ approximation.

H.2 Monte Carlo audit on HotpotQA

We audit the exact reduced score on HotpotQA, where max-step prefixes can be resumed from the stopped state. We sampled 100 naturally max-step prefixes and generated up to ten continuation rollouts per prefix under the same ReAct harness. Continuation branches are used only to estimate $\widehat{q}^{\mathrm{MC}}_{Z,i}$; the scored forecast stream $F_{i,1:Z_i}$ is frozen from the original source-prefix predictor record.

After excluding prefixes with fewer than five valid resolved non-parse continuations, the audit retains 87 prefixes. The retained prefixes have 8.92 valid resolved branches on average, and the mean number of parse-error branches excluded per prefix is 0.115. The canonical source-prefix join ensures that the scored prefix forecast is identical across continuation branches:

$$\max_{i,m,t} \left| F_{it}^{(m)} - F_{it}^{\mathrm{source}} \right| = 0.$$

For a censored prefix i, the simple and exact-MC log scores are

$$\mathrm{TPS}_i^{\mathrm{cen,simple}} = \sum_{t=1}^{Z_i} w_{it}\, S_{\log}(F_{it}, 0), \tag{16}$$

$$\mathrm{TPS}_i^{\mathrm{cen,exact\text{-}MC}} = \sum_{t=1}^{Z_i} w_{it} \left[\widehat{q}^{\mathrm{MC}}_{Z,i} S_{\log}(F_{it}, 1) + \left(1 - \widehat{q}^{\mathrm{MC}}_{Z,i}\right) S_{\log}(F_{it}, 0) \right]. \tag{17}$$

The paired correction has the closed form

$$\Delta_i = \mathrm{TPS}_i^{\mathrm{cen,exact\text{-}MC}} - \mathrm{TPS}_i^{\mathrm{cen,simple}} = \widehat{q}^{\mathrm{MC}}_{Z,i} \sum_{t=1}^{Z_i} w_{it} \log \frac{F_{it}}{1 - F_{it}}. \tag{18}$$

Thus the sign of the exact-minus-simple correction is governed by the weighted prefix log-odds.

Conditional-projection identity. The exact-MC score should equal the branch-average complete-prefix score:

$$\mathrm{TPS}_i^{\mathrm{cen,exact\text{-}MC}} = \frac{1}{M_i^{\mathrm{valid}}} \sum_{m=1}^{M_i^{\mathrm{valid}}} \sum_{t=1}^{Z_i} w_{it}\, S_{\log}(F_{it}, Y_i^{(m)}).$$

This identity holds numerically in the audit. On the primary predictor,

$$\max_i \left| \mathrm{TPS}_i^{\mathrm{cen,exact\text{-}MC}} - \overline{\mathrm{TPS}}_i^{\mathrm{branch}} \right| = 1.1 \times 10^{-16},$$

and the maximum error is at most 9×10^{-16} across all retained reported predictors. This verifies numerically that the implemented score realizes the conditional projection used in the censored-score derivation.

Estimated q_Z distribution. The estimated continuation-success probabilities are heterogeneous across HotpotQA max-step prefixes. The mean is $\widehat{q}_Z^{\mathrm{MC}} = 0.240$, with 95% bootstrap CI $[0.152, 0.327]$. The median is 0, 67.8% of retained prefixes have $\widehat{q}_Z^{\mathrm{MC}} = 0$, and 23.0% have $\widehat{q}_Z^{\mathrm{MC}} > 0.5$. Thus the stopped-prefix distribution contains both deeply stuck prefixes and a recoverable upper tail.

Diagnostic	Value
Mean $\widehat{q}_Z^{\mathrm{MC}}$	$0.240 \quad [0.152,\ 0.327]$
Median $\widehat{q}_Z^{\mathrm{MC}}$	0.000
Fraction with $\widehat{q}_Z^{\mathrm{MC}} = 0$	67.8%
Fraction with $\widehat{q}_Z^{\mathrm{MC}} > 0.5$	23.0%

Table 8: Estimated continuation-success probabilities on 87 retained HotpotQA max-step prefixes. Brackets give the 95% bootstrap CI for the mean. The distribution has a large mass at zero and a recoverable upper tail.

Primary score comparison. Table 9 reports the primary comparison for Platt-cross-fitted verbal confidence with linear-front weights. The exact-MC score is lower than the simple approximation by 0.105 nats, with a 95% bootstrap CI excluding zero. By Eq. (18), this direction is expected because the calibrated HotpotQA max-step prefixes mostly have weighted prefix log-odds below zero.

Quantity	Value	95% CI
$\widehat{\mathrm{TPS}}^{\mathrm{cen,simple}}$	-0.469	$[-0.485,\ -0.457]$
$\widehat{\mathrm{TPS}}^{\mathrm{cen,exact\text{-}MC}}$	-0.574	$[-0.619,\ -0.530]$
$\Delta_{\mathrm{exact-simple}}$	-0.105	$[-0.148,\ -0.065]$

Table 9: Exact-MC versus simple censored log score on HotpotQA max-step prefixes. Primary predictor: Platt-cross-fitted verbal confidence, linear-front weights. Bootstrap intervals use 1000 prefix-level resamples.

How accurate must q_Z be? Across the 87 retained prefixes, the slope c_i from Eq. (15) has mean -0.527, and the middle 90% of prefixes lie in $[-0.695, -0.238]$. To keep the score within 0.05 nats, q_Z can therefore be off by about 0.095 for a typical prefix and by about 0.072 at the 95th percentile of sensitivity. Without any reruns, letting each prefix's q_Z range over $[0, 1]$ bounds the dataset-level score to $[-0.998, -0.467]$; the estimate from reruns, -0.574, falls inside this range.

How many reruns are needed? We repeat the analysis using only $M \in \{2, 5, 7, 10\}$ of the cached continuations per prefix, redrawing the selection 5,000 times and using 2,000 nested bootstrap resamples for intervals. To compare budgets fairly, every budget uses the same 54 prefixes that have ten valid continuations. The exact score is below the simple score in every repetition at every budget, and going from $M = 2$ to $M = 10$ narrows the interval only from 0.125 to 0.119 nats. Most of the uncertainty therefore comes from differences between prefixes, not from the number of reruns.

Estimating q_Z for a single prefix is harder. Only four of the 54 prefixes have both successful and failed continuations; the other 50 had the same outcome in all ten reruns. For those four, the average error in q_Z falls from 0.217 to 0.112 to 0.070 as M goes from 2 to 5 to 7, and the resulting error in the exact score falls from 0.148 to 0.074 to 0.048 nats. These errors are measured against the ten-rerun estimate, not the unknown true value. If we instead use every prefix with enough valid reruns at each budget ($87, 87, 76, 54$ prefixes), the exact score is still below the simple score in every repetition, but differences in magnitude then partly reflect which prefixes are included. In short, a few reruns suffice for a dataset-level conclusion, but not for a precise estimate for an individual prefix.

Predictor-level robustness. Table 10 reports the exact-minus-simple correction across the predictor pool. The same negative correction appears for every Platt-calibrated predictor under both linear-front and uniform weights, while raw saturated predictors generally show the opposite sign. This is exactly the behavior predicted by Eq. (18): the correction is controlled by the weighted prefix log-odds, not by the identity of the predictor.

Unresolved-branch sensitivity. The audit is stable to adversarial handling of unresolved continuation branches. Treating the unresolved branches as failures gives $\Delta_{\mathrm{exact-simple}} = -0.101$ nats; treating

Predictor	Weights	Simple	Exact-MC	$\Delta_{\text{exact}-\text{simple}}$
Verbal confidence, Platt	linear-front	-0.469	-0.574	-0.105 [-0.148, -0.065]
Completion entropy, Platt	linear-front	-0.437	-0.598	-0.162 [-0.224, -0.100]
Completion token prob., Platt	linear-front	-0.452	-0.598	-0.145 [-0.209, -0.093]
Action-span entropy, Platt	linear-front	-0.523	-0.619	-0.096 [-0.134, -0.059]
Action-span token prob., Platt	linear-front	-0.525	-0.620	-0.095 [-0.135, -0.062]
Base-rate constant	linear-front	-0.596	-0.645	-0.049 [-0.066, -0.033]
Verbal confidence, Platt	uniform	-0.427	-0.558	-0.132 [-0.186, -0.082]
Completion entropy, Platt	uniform	-0.433	-0.600	-0.166 [-0.238, -0.104]
Completion token prob., Platt	uniform	-0.448	-0.599	-0.150 [-0.217, -0.099]
Action-span entropy, Platt	uniform	-0.477	-0.604	-0.127 [-0.171, -0.080]
Action-span token prob., Platt	uniform	-0.480	-0.605	-0.125 [-0.175, -0.080]
Verbal confidence, raw	linear-front	-1.925	-1.287	$+0.638$ [$+0.337$, $+1.018$]
Completion entropy, raw	linear-front	-2.054	-1.591	$+0.463$ [$+0.307$, $+0.632$]
Completion token prob., raw	linear-front	-2.465	-1.888	$+0.577$ [$+0.376$, $+0.786$]
Action-span entropy, raw	linear-front	-3.487	-2.685	$+0.801$ [$+0.521$, $+1.125$]
Action-span token prob., raw	linear-front	-4.190	-3.213	$+0.977$ [$+0.629$, $+1.367$]

Table 10: Predictor-level exact-q_Z audit on 87 retained HotpotQA max-step prefixes. Scores are log-TPS rewards in nats. The exact-minus-simple correction is negative for all Platt-calibrated reported predictors under both weight schedules, and positive for raw saturated streams under linear-front weights, consistent with the prefix log-odds formula in Eq. (18). The conditional-projection identity holds to numerical precision across all rows ($\max_i |\mathrm{TPS}_i^{\text{cen,exact-MC}} - \overline{\mathrm{TPS}}_i^{\text{branch}}| \le 9 \times 10^{-16}$).

them as successes gives $\Delta_{\text{exact}-\text{simple}} = -0.158$ nats. Both sensitivity analyses preserve the sign and keep the confidence interval away from zero.

The exact q_Z-weighted censored score is operationally computable on real stopped prefixes. On HotpotQA, Monte Carlo continuation yields nontrivial stopped-prefix success probabilities, the exact score differs materially from the simple $q_Z \approx 0$ approximation, the correction direction follows the prefix log-odds algebra, and the conditional-projection identity holds to numerical precision.

I Score-Family and Weight-Schedule Sensitivity

This appendix reports score-family and weight-schedule sensitivity for the artificial-censoring validation in Section 6.2 and the natural-censoring WebShop analysis in Section 6.3. The primary experiments use log score with linear-front weights unless otherwise stated. Here we vary both the trajectory-weight schedule and the binary score family.

Weight schedules. All schedules are normalized over the full trajectory length T:

$$\sum_{t=1}^{T} w_t = 1.$$

For censored prefixes, we use the same full-trajectory weights and truncate the sum at the observed stopping step Z; weights are not renormalized over the observed prefix. Thus

$$\sum_{t=1}^{Z} w_t \le 1,$$

with strict inequality when the trajectory is censored before the full trajectory length.

As a practical convention, uniform weights are appropriate when every prefix is equally decision-relevant; linear-front or exponential-front weights are appropriate for early-warning, deferral, or intervention settings; and linear-back weights are appropriate when late-stage confidence is the primary object. The weight schedule should be chosen before scoring and reported as part of the evaluation protocol.

We evaluate four schedules:

- **Linear-front**:
$$w_t = \frac{2(T-t+1)}{T(T+1)}.$$
This is the primary schedule used in the main experiments. It gives larger weight to early prefixes, where miscalibrated uncertainty can affect subsequent planning, tool use, reflection, or deferral.
- **Uniform**:
$$w_t = \frac{1}{T}.$$
This treats all prefix positions equally.
- **Exponential-front**:
$$w_t = \frac{2^{-(t-1)}}{\sum_{s=1}^{T} 2^{-(s-1)}} = \frac{2^{-(t-1)}}{2(1-2^{-T})}.$$
This is a more aggressively front-loaded schedule: each raw step weight is half the previous one.
- **Linear-back**:
$$w_t = \frac{2t}{T(T+1)}.$$
This is the mirror of linear-front and gives larger weight to later prefixes.

Score families. We report three score families: log, Brier, and Beta(2,4). The log score is the primary censored row. Brier and Beta(2,4) are robustness rows under the same failure-side $\mathrm{TPS}^{\mathrm{cen,simple}}$ approximation. They should be read as operational sensitivity checks, not as separate theorem-level censored propriety claims for the simple $q_Z \approx 0$ approximation.

I.1 Artificial-Censoring Validation

On complete StrategyQA, Tau2-Bench, and HotpotQA trajectories, we artificially hide outcomes at length-stratified rates $r \in \{0.25, 0.50, 0.75\}$, score the observed prefixes with $\mathrm{TPS}^{\mathrm{cen,simple}}$, and compare against the complete-data trajectory score. Because the true outcome is available for every trajectory, the estimator's behavior can be checked exactly against closed-form predictions rather than treated as a black-box approximation.

For each artificially censored trajectory, the score change decomposes into two analytic terms. The *prefix-swap* term replaces the success branch on the observed prefix with the failure branch; for the log score its stepwise sign is $\log(1-F_t) - \log F_t$, which flips at $F_t = 0.5$. The *tail-omission* term drops the unobserved suffix and is always non-negative. Across 69,336 evaluated combinations of trajectories, censoring rates, score families, and weight schedules, the closed-form decomposition matches the directly computed $\mathrm{TPS}^{\mathrm{cen,simple}} - \mathrm{TPS}^{\mathrm{complete}}$ difference to numerical precision, with maximum absolute error 4.4×10^{-16}.

At the dataset level, the same decomposition explains the observed signs and magnitudes (Table 11). Under log score with linear-front weights at $r = 0.75$, StrategyQA is in a high-confidence/high-success regime and yields $\Delta_{\mathrm{mean}} = -0.675$ nats. Tau2 has calibrated forecasts concentrated below 0.5, so the prefix-swap contribution turns positive, yielding $+0.303$ nats. HotpotQA is the intermediate case: the negative successful-trajectory prefix-swap is partly offset by positive failure-side and tail-omission terms, yielding $+0.056$ nats. The mean shift scales approximately linearly with censoring rate across all three datasets.

For the original continuation-success target, the theorem-backed censored score is the exact q_Z-weighted reduction in Section 4.3. The simple censored score used here is the operational $q_Z \approx 0$ approximation from Section 4.4; Proposition C.3 identifies its pseudo-label target, and the artificial-censoring experiment characterizes its bias relative to complete-data scoring. Brier and Beta(2,4) are reported as robustness checks under the same failure-side approximation, not as separate theorem-level censored propriety claims. Thus $\mathrm{TPS}^{\mathrm{cen,simple}}$ is not a uniformly pessimistic lower bound on the complete-data score. Its shift can be positive or negative, but the direction is analytically decomposable.

Table 11 reports the artificial-censoring robustness sweep at $r = 0.75$, varying both the score family and the trajectory-weight schedule.

The weight schedule changes the magnitude, but not the main regime-level pattern. Front-loaded schedules amplify the effect of early censored-prefix replacement; linear-back shifts weight toward later prefixes and therefore usually attenuates the shift when early prefixes dominate the correction. StrategyQA remains negative across all schedules, Tau2-Bench remains positive across all schedules, and HotpotQA remains the boundary case: log-score rows stay positive, while bounded or asymmetric rows near zero can change sign under different schedules.

Here $\mathrm{TPS}^{\mathrm{complete}}$ is the complete-data mean TPS at $r = 0$, and Δ_{mean} is the mean shift under $\mathrm{TPS}^{\mathrm{cen,simple}}$ relative to the complete-data score. The final column reports the mean shift separately on failure and success trajectories.

I.2 Natural Censoring on WebShop

The WebShop working sample contains 308 trajectories after excluding parse-error terminations as informative censoring: 163 completed trajectories with observed outcomes and 145 administratively censored max-step prefixes. The administrative-censoring rate within the working sample is 47.08%. The primary WebShop configuration in Section 6.3 is Platt-calibrated verbal confidence scored with log score and linear-front weights:

$$\widehat{\mathrm{TPS}}_{\mathrm{comp}} = -0.8156, \qquad \widehat{\mathrm{TPS}}_{\text{cen-ext}} = -0.6570, \qquad \Delta_{\mathrm{practice}} = +0.1586,$$

with 95% bootstrap CI $[+0.1334, +0.1879]$.

Table 12 reports the full predictor–score-family–weight-schedule sweep in compressed form. The four underconfident or near-base-rate Platt predictors have positive $\Delta_{\mathrm{practice}}$ across all score families and schedules. The completion-token probability predictor is the only non-reference predictor with $\bar{F} > 0.5$, and it is negative across all twelve score-family–weight cells.

Overconfident predictor block. The completion-token probability predictor is separated because it is the only non-reference predictor with calibrated mean forecast above 0.5. Its $\Delta_{\mathrm{practice}}$ is negative for every score family and weight schedule, and its magnitude changes little across schedules. This matches the artificial-censoring decomposition: under the failure-side branch, overconfident censored-prefix forecasts are penalized.

Sign stability across predictors. Table 14 summarizes the sign of $\Delta_{\mathrm{practice}}$ across the five non-reference predictors. Across the $5 \times 3 \times 4 = 60$ non-reference predictor–score–weight cells, 48 are positive and 12 are negative. All 12 negative cells correspond to completion-token probability, the only predictor in this sweep with $\bar{F} > 0.5$.

Margin sensitivity relative to the base-rate reference. Table 15 reports the log-score margin of verbal confidence over the base-rate reference. The censored extension reduces the margin under every weight schedule. Under exponential-front weights, the margin changes sign.

J Beta-family Scoring Rules

This appendix records the strict-propriety argument for the beta-family score used in Section 3, describes its boundary behavior, and explains how the parameters (α, β) act as cost-shaping choices. It also notes how the same score can be used as a calibration loss under a fixed evaluation law.

Throughout this appendix, $S_{\alpha,\beta}$ denotes the per-step binary score, while TPS denotes its trajectory-level weighted lift.

J.1 Strict propriety of the beta family

Recall that the beta-family score is

$$S_{\alpha,\beta}(p, 1) = -\int_p^1 c^{\alpha-1}(1-c)^{\beta}\, dc, \qquad S_{\alpha,\beta}(p, 0) = -\int_0^p c^{\alpha}(1-c)^{\beta-1}\, dc.$$

Dataset	Score	Weight schedule	$\mathrm{TPS}^{\mathrm{complete}}$	Δ_{mean}	$(\Delta_{Y=0},\ \Delta_{Y=1})$
StrategyQA	log	linear-front	-0.4186	-0.6746	$(+0.319,\ -0.862)$
StrategyQA	Brier	linear-front	-0.1283	-0.2862	$(+0.117,\ -0.362)$
StrategyQA	Beta(2,4)	linear-front	-0.0026	-0.0073	$(+0.003,\ -0.009)$
StrategyQA	log	uniform	-0.4169	-0.4197	$(+0.544,\ -0.601)$
StrategyQA	Brier	uniform	-0.1278	-0.1905	$(+0.202,\ -0.264)$
StrategyQA	Beta(2,4)	uniform	-0.0026	-0.0051	$(+0.005,\ -0.007)$
StrategyQA	log	exponential-front	-0.4203	-0.7657	$(+0.198,\ -0.947)$
StrategyQA	Brier	exponential-front	-0.1288	-0.3232	$(+0.071,\ -0.397)$
StrategyQA	Beta(2,4)	exponential-front	-0.0026	-0.0083	$(+0.002,\ -0.010)$
StrategyQA	log	linear-back	-0.4147	-0.2306	$(+0.820,\ -0.429)$
StrategyQA	Brier	linear-back	-0.1275	-0.1126	$(+0.300,\ -0.190)$
StrategyQA	Beta(2,4)	linear-back	-0.0026	-0.0030	$(+0.007,\ -0.005)$
Tau2-Bench	log	linear-front	-0.7210	$+0.3032$	$(+0.061,\ +0.591)$
Tau2-Bench	Brier	linear-front	-0.2617	$+0.1309$	$(+0.017,\ +0.266)$
Tau2-Bench	Beta(2,4)	linear-front	-0.0082	$+0.0030$	$(+0.001,\ +0.006)$
Tau2-Bench	log	uniform	-0.7185	$+0.3598$	$(+0.117,\ +0.649)$
Tau2-Bench	Brier	uniform	-0.2607	$+0.1468$	$(+0.033,\ +0.283)$
Tau2-Bench	Beta(2,4)	uniform	-0.0081	$+0.0037$	$(+0.002,\ +0.006)$
Tau2-Bench	log	exponential-front	-0.7480	$+0.3078$	$(+0.007,\ +0.666)$
Tau2-Bench	Brier	exponential-front	-0.2717	$+0.1379$	$(+0.002,\ +0.300)$
Tau2-Bench	Beta(2,4)	exponential-front	-0.0087	$+0.0034$	$(+0.000,\ +0.007)$
Tau2-Bench	log	linear-back	-0.7030	$+0.3924$	$(+0.180,\ +0.646)$
Tau2-Bench	Brier	linear-back	-0.2549	$+0.1536$	$(+0.051,\ +0.275)$
Tau2-Bench	Beta(2,4)	linear-back	-0.0078	$+0.0040$	$(+0.003,\ +0.006)$
HotpotQA	log	linear-front	-0.6771	$+0.0558$	$(+0.196,\ -0.044)$
HotpotQA	Brier	linear-front	-0.2420	$+0.0070$	$(+0.077,\ -0.043)$
HotpotQA	Beta(2,4)	linear-front	-0.0065	-0.0020	$(+0.003,\ -0.006)$
HotpotQA	log	uniform	-0.6767	$+0.1703$	$(+0.310,\ +0.070)$
HotpotQA	Brier	uniform	-0.2418	$+0.0519$	$(+0.122,\ +0.002)$
HotpotQA	Beta(2,4)	uniform	-0.0065	-0.0003	$(+0.005,\ -0.004)$
HotpotQA	log	exponential-front	-0.6774	$+0.0225$	$(+0.161,\ -0.077)$
HotpotQA	Brier	exponential-front	-0.2422	-0.0059	$(+0.064,\ -0.055)$
HotpotQA	Beta(2,4)	exponential-front	-0.0065	-0.0025	$(+0.002,\ -0.006)$
HotpotQA	log	linear-back	-0.6765	$+0.2725$	$(+0.434,\ +0.157)$
HotpotQA	Brier	linear-back	-0.2418	$+0.0907$	$(+0.171,\ +0.034)$
HotpotQA	Beta(2,4)	linear-back	-0.0065	$+0.0013$	$(+0.006,\ -0.002)$

Table 11: Artificial-censoring robustness at $r = 0.75$. The table reports the complete-data TPS and the mean shift under $\mathrm{TPS}^{\mathrm{cen,simple}}$, with the shift also reported separately on failure and success trajectories. StrategyQA is negative across all rows, Tau2-Bench is positive across all rows, and HotpotQA is an intermediate regime in which bounded or asymmetric rows near zero can change sign.

Predictor	Weight schedule	$\Delta\mathrm{TPS}_{\log}$	$\Delta\mathrm{TPS}_{\mathrm{Brier}}$	$\Delta\mathrm{TPS}_{\mathrm{Beta}(2,4)}$
Verbal confidence	linear-front	$+0.1586$	$+0.0766$	$+0.00060$
Verbal confidence	uniform	$+0.1559$	$+0.0736$	$+0.00072$
Verbal confidence	exponential-front	$+0.1520$	$+0.0735$	$+0.00047$
Verbal confidence	linear-back	$+0.1463$	$+0.0707$	$+0.00036$
Completion entropy confidence	linear-front	$+0.1585$	$+0.0776$	$+0.00054$
Completion entropy confidence	uniform	$+0.1573$	$+0.0769$	$+0.00054$
Completion entropy confidence	exponential-front	$+0.1567$	$+0.0769$	$+0.00047$
Completion entropy confidence	linear-back	$+0.1601$	$+0.0783$	$+0.00058$
Action-span entropy confidence	linear-front	$+0.1623$	$+0.0797$	$+0.00057$
Action-span entropy confidence	uniform	$+0.1623$	$+0.0797$	$+0.00057$
Action-span entropy confidence	exponential-front	$+0.1623$	$+0.0797$	$+0.00057$
Action-span entropy confidence	linear-back	$+0.1623$	$+0.0797$	$+0.00057$
Action-span token-prob. confidence	linear-front	$+0.1623$	$+0.0797$	$+0.00057$
Action-span token-prob. confidence	uniform	$+0.1618$	$+0.0793$	$+0.00056$
Action-span token-prob. confidence	exponential-front	$+0.1623$	$+0.0797$	$+0.00057$
Action-span token-prob. confidence	linear-back	$+0.1623$	$+0.0797$	$+0.00057$
Completion token probability	linear-front	-2.0686	-0.1175	-0.00241
Completion token probability	uniform	-2.0700	-0.1182	-0.00243
Completion token probability	exponential-front	-2.0686	-0.1175	-0.00241
Completion token probability	linear-back	-2.0686	-0.1175	-0.00241
Base-rate reference	linear-front	$+0.1623$	$+0.0797$	$+0.00057$
Base-rate reference	uniform	$+0.1623$	$+0.0797$	$+0.00057$
Base-rate reference	exponential-front	$+0.1623$	$+0.0797$	$+0.00057$
Base-rate reference	linear-back	$+0.1623$	$+0.0797$	$+0.00057$

Table 12: WebShop natural-censoring robustness across predictors, score families, and weight schedules. Entries are $\Delta_{\mathrm{practice}} = \widehat{\mathrm{TPS}}_{\text{cen-ext}} - \widehat{\mathrm{TPS}}_{\mathrm{comp}}$. Positive values mean the censored extension gives a higher score than complete-only evaluation; negative values mean it gives a lower score.

Score	Weight schedule	$\widehat{\mathrm{TPS}}_{\mathrm{comp}}$	$\widehat{\mathrm{TPS}}_{\text{cen-ext}}$	$\Delta_{\mathrm{practice}}$	95% CI
log	linear-front	-2.5255	-4.5941	-2.0686	$[-2.7092, -1.3721]$
log	uniform	-2.5233	-4.5932	-2.0700	$[-2.7496, -1.3661]$
log	exponential-front	-2.5255	-4.5941	-2.0686	$[-2.8006, -1.4609]$
log	linear-back	-2.5255	-4.5941	-2.0686	$[-2.7303, -1.4200]$
Brier	linear-front	-0.3091	-0.4266	-0.1175	$[-0.1590, -0.0750]$
Brier	uniform	-0.3081	-0.4263	-0.1182	$[-0.1615, -0.0749]$
Brier	exponential-front	-0.3091	-0.4266	-0.1175	$[-0.1577, -0.0734]$
Brier	linear-back	-0.3091	-0.4266	-0.1175	$[-0.1622, -0.0761]$
Beta(2,4)	linear-front	-0.00656	-0.00897	-0.00241	$[-0.00304, -0.00182]$
Beta(2,4)	uniform	-0.00651	-0.00894	-0.00243	$[-0.00310, -0.00183]$
Beta(2,4)	exponential-front	-0.00656	-0.00897	-0.00241	$[-0.00307, -0.00183]$
Beta(2,4)	linear-back	-0.00656	-0.00897	-0.00241	$[-0.00306, -0.00178]$

Table 13: WebShop natural-censoring robustness for Platt-calibrated completion-token probability. This predictor has $\bar{F} \approx 0.70$ and is the only non-reference predictor with uniformly negative $\Delta_{\mathrm{practice}}$ across score families and weight schedules.

Predictor	$\bar{F}$	linear-front	uniform	exponential-front	linear-back
Verbal confidence (Platt)	0.38	+	+	+	+
Completion entropy conf. (Platt)	0.38	+	+	+	+
Action-span entropy conf. (Platt)	0.38	+	+	+	+
Action-span token prob. (Platt)	0.38	+	+	+	+
Completion token prob. (Platt)	0.70	−	−	−	−

Table 14: Sign of Δ_{practice} for non-reference predictors on WebShop. Each sign summarizes the corresponding score-family rows under that weight schedule. The sign is determined by the predictor's confidence regime rather than by the choice of score family or weight schedule.

Weight schedule	$\widehat{\text{TPS}}_{\text{comp}} - \widehat{\text{TPS}}_{\text{base}}$	$\widehat{\text{TPS}}_{\text{cen-ext}} - \widehat{\text{TPS}}_{\text{base}}$	Shrinkage	Rank change?
linear-front	+0.0123	+0.0086	−30%	No
uniform	+0.0301	+0.0237	−21%	No
exponential-front	+0.0079	−0.0025	rank flip	Yes
linear-back	+0.0476	+0.0316	−34%	No

Table 15: Verbal-confidence margin over the base-rate reference under log score. The censored extension reduces the margin under every weight schedule; under exponential-front weights, the margin changes sign.

Strict propriety. The score is defined by the Schervish–Buja threshold-mixture construction with mixing measure

$$\nu(dc) = c^{\alpha-1}(1-c)^{\beta-1}\,dc.$$

For $\alpha > 0$ and $\beta > 0$, this density is strictly positive on every open subinterval of $(0,1)$. Hence the mixing measure has full support on $(0,1)$. By Schervish's characterization of binary proper scores, a threshold-mixture score is strictly proper if and only if the mixing measure assigns positive mass to every open subinterval of $(0,1)$. Therefore $S_{\alpha,\beta}$ is strictly proper for all $\alpha, \beta > 0$.

The Brier score corresponds to $\alpha = \beta = 1$, up to positive affine equivalence:

$$S_{1,1}(p,1) = -\frac{(1-p)^2}{2}, \qquad S_{1,1}(p,0) = -\frac{p^2}{2}.$$

Thus maximizing $S_{1,1}$ is equivalent to minimizing the usual Brier loss.

The logarithmic endpoint is obtained as the limiting endpoint $\alpha = \beta \to 0$. We treat it as the standard binary log score,

$$S_{\log}(p,1) = \log p, \qquad S_{\log}(p,0) = \log(1-p),$$

which is strictly proper by direct verification. If $Y \mid \mathcal{H}_t \sim \text{Bernoulli}(q_t)$, then the conditional expected log score is

$$q_t \log p + (1-q_t)\log(1-p).$$

Its derivative is

$$\frac{q_t}{p} - \frac{1-q_t}{1-p},$$

which vanishes uniquely at $p = q_t$, and its second derivative is negative on $(0,1)$. Hence the log score is strictly proper. □

J.2 Boundary behavior

The log score is unbounded at the wrong boundary:

$$S_{\log}(p,1) \to -\infty \quad \text{as } p \to 0, \qquad S_{\log}(p,0) \to -\infty \quad \text{as } p \to 1.$$

In implementations, we therefore clip reported probabilities to $[\varepsilon, 1-\varepsilon]$, with $\varepsilon = 10^{-6}$, before computing any log score.

Interior beta-family members are bounded on $[0, 1]$. The two wrong-boundary floors are finite:

$$S_{\alpha,\beta}(0, 1) = -\int_0^1 c^{\alpha-1}(1-c)^{\beta}\, dc = -B(\alpha, \beta+1),$$

and

$$S_{\alpha,\beta}(1, 0) = -\int_0^1 c^{\alpha}(1-c)^{\beta-1}\, dc = -B(\alpha+1, \beta),$$

where $B(\cdot, \cdot)$ is the beta function. Thus, unlike the log score, interior beta-family members cap the penalty for near-certain wrong forecasts.

This distinction matters empirically because agentic confidence streams can be saturated. Verbal-confidence predictors, for example, often cluster near 0.95 or 1.00. Log score penalizes such saturation sharply when the trajectory fails, whereas bounded beta-family members saturate at a finite penalty. This is why the experiments report log as the canonical unbounded default, Brier as the bounded symmetric baseline, and Beta(2,4) as a bounded asymmetric sensitivity check.

The boundedness of interior beta-family members also explains the scope of the approximate-propriety statement in Remark C.5. The contrast

$$D(p) = S_{\alpha,\beta}(p, 1) - S_{\alpha,\beta}(p, 0)$$

is bounded for $\alpha, \beta > 0$, but not for the log endpoint. Hence finite plug-in error bounds for misspecified censored weights apply cleanly to interior beta-family members and require additional clipping or boundedness conditions for log score.

J.3 Cost-shaping interpretation

The parameters (α, β) do not determine whether the score is proper. Strict propriety follows from full support of the mixing measure. Instead, (α, β) determine how the proper score weights different threshold mistakes.

In the Schervish threshold-mixture representation, each threshold $c \in (0, 1)$ can be viewed as a cost-sensitive threshold rule. The rule penalizes a forecast according to whether the reported probability falls on the wrong side of threshold c. The beta-family mixing measure

$$\nu(dc) \propto c^{\alpha-1}(1-c)^{\beta-1}\, dc$$

controls how much weight is assigned to each threshold region.

Smaller α relative to β shifts mass toward lower thresholds. This places more weight on mistakes associated with reporting success probabilities above thresholds that the eventual failed trajectory does not justify. In agentic settings, this is the overconfident-success regime: the agent reports a high probability of eventual success, but the trajectory fails. Such errors can delay reflection, deferral, or human handoff.

Larger α relative to β shifts weight toward higher thresholds and can emphasize underconfidence on successful trajectories. This may be appropriate when unnecessary interventions are costly and the system should be encouraged to proceed when success is likely. The Brier score, corresponding to $\alpha = \beta = 1$ up to positive affine equivalence, is the symmetric bounded case.

Choosing a beta score from a decision threshold. Suppose a customer-service agent hands the conversation to a human when its success probability drops below 0.30. Beta(4,8) has its mode at 0.30, so it puts the most weight on probability errors near that handoff point. A research agent that runs an expensive verification step when its success probability exceeds 0.80 might instead use Beta(5,2), whose mode is 0.80. In general, for a threshold $\tau \in (0, 1)$, choose $\alpha, \beta > 1$ with

$$\frac{\alpha - 1}{\alpha + \beta - 2} = \tau.$$

Larger values of $\alpha + \beta$ concentrate the score more tightly around τ; choose them according to how sharply the decision depends on the threshold, and fix them before comparing systems. When no threshold is privileged, use log as the main score and Brier, which weights all thresholds equally, as a check. Reporting several scores is fine if they are chosen in advance, but not if only the most favorable one is reported.

Choosing step weights. Step weights can be chosen in the same spirit, as $w_t \propto v_t$, where $v_t > 0$ expresses how valuable an accurate forecast is at step t. These values can come from domain knowledge or from a separate pilot set. For example, if a pilot shows that recovery from an error becomes rare after step 3, accurate forecasts in the first three steps are the ones that can still trigger a useful intervention, so they deserve more weight. Uniform weights treat all steps equally; front-loaded weights suit early intervention and back-loaded weights suit a final check. As with the score, the weights must be fixed before comparing systems.

Beta(2,4) provides a bounded asymmetric instance of the beta family. Together with log and Brier, it lets us compare an unbounded default, a bounded symmetric score, and a bounded cost-shaped score.

J.4 Proper scoring rules as calibration losses

A strictly proper scoring rule can also be used as a calibration-aware loss. Under the reward orientation used in this paper, the corresponding trajectory-level loss is the negated score:

$$\mathcal{L}_{\mathrm{traj},\alpha,\beta}(F_{1:T}, Y) = -\sum_{t=1}^{T} w_t \, S_{\alpha,\beta}(F_t, Y).$$

For complete trajectories, minimizing this expected loss is equivalent to maximizing the expected TPS. By Theorem 4.1, the unique population optimum is

$$F_t = q_t = \mathbb{P}^{\pi}(Y = 1 \mid \mathcal{H}_t) \quad \text{for every } t.$$

Thus the same mathematical object can be used either as an evaluator of an existing uncertainty stream or as a loss for training a confidence head, post-hoc calibrator, or uncertainty probe under a fixed policy.

This differs from optimizing trajectory ECE. An optimizer minimizing ECE is rewarded for marginal calibration within bins, not for the resolution that proper scoring rules require. It can therefore fail to recover the full prefix-conditioned probability process even when binwise calibration error is small.

The (α, β) parameters provide a risk-preference interface for the loss:

- Overconfident success predictions on failed trajectories ($p \to 1, Y = 0$) can delay reflection, deferral, or human handoff. A bounded asymmetric member with $\alpha < \beta$ can emphasize this regime while avoiding the unbounded log penalty.
- Excessive underconfidence on successful trajectories ($p \to 0, Y = 1$) can trigger unnecessary interventions. A choice with $\alpha > \beta$ can emphasize this regime.
- When the two directions are treated symmetrically, Brier ($\alpha = \beta = 1$) and log are natural defaults, with Brier bounded and log unbounded.

Distinction from policy optimization. Using $-\mathrm{TPS}$ as a calibration loss is not the same as using it as a policy-optimization or RLHF reward. The propriety theorem fixes the evaluation law and scores forecasts under that law. If the policy itself is optimized against the score, it may change the distribution of histories being evaluated, conflating policy improvement with calibration. We therefore use $-\mathrm{TPS}$ as a loss for confidence heads, post-hoc calibrators, or uncertainty probes under a fixed policy, not as a standalone RLHF reward.

Training on TPS. Propriety holds for the distribution of trajectories being evaluated. A model trained to maximize TPS can therefore learn to report well-calibrated probabilities on trajectories like those it was trained on without those probabilities carrying over to new tasks, so it should be evaluated on held-out tasks and under distribution shift. TPS also rewards truthfulness, not success: an agent that correctly predicts it will fail scores well. Task success must be rewarded separately, and a good TPS score alone does not justify reducing human oversight.

Censored trajectories. For complete trajectories, $-\mathrm{TPS}_{\alpha,\beta}$ is a valid calibration loss for any strictly proper beta-family member. For censored trajectories, the exact reduced censored loss requires the continuation-success weight q_Z:

$$q_Z = \mathbb{P}^{\pi}(Y = 1 \mid \mathcal{H}_Z).$$

When q_Z is available or consistently estimated, the exact reduced score in Eq. (4) can be negated and used as the corresponding censored calibration loss. When q_Z is unavailable, the simple failure-side

approximation $q_Z \approx 0$ gives $-\mathrm{TPS}^{\mathrm{cen,simple}}_{\alpha,\beta}$. As Proposition C.3 shows, this simple loss is proper for the pseudo-label target

$$m_t = \mathbb{P}^\pi(\Delta = 1, Y = 1 \mid \mathcal{H}_t),$$

not for the original continuation-success target q_t, unless the missing success mass is zero. Thus $\mathrm{TPS}^{\mathrm{cen,simple}}$ is an operational approximation, while $\mathrm{TPS}^{\mathrm{cen,exact}}$ is the strictly proper censored score.

K Implementation Details

Model and inference. All trajectories were collected from `google/gemma-4-31b-it` accessed through OpenRouter [29]. Token-level log-probabilities at every step are the raw materials for the completion-token and action-span predictors. Table 16 summarizes decoding settings; all values are shared across datasets.

Hyperparameter	StrategyQA / Tau2 / HotpotQA / WebShop
`temperature`	0.2
`top_p`	1.0
`top_logprobs`	5
`max_completion_tokens`	64,000
`reasoning`	enabled

Table 16: Decoding hyperparameters.

Agent harness. All benchmarks used an AUQ-UAM-style ReAct harness [30] eliciting four structured fields at every step: `<think>`, `<action>`, `<confidence>`, and `<explanation>`. The `<confidence>` field is the verbal-confidence predictor. Up to two repair attempts were allowed per step before a parse-error termination was recorded.

Missing predictor values. We do not impute missing per-step forecasts. Malformed or missing structured confidence fields are handled by the harness retry logic; if parsing still fails, the episode is assigned stop reason `parse_error` and is excluded before predictor construction. For residual predictor-level missingness, the raw predictor may record a missing value, but the evaluator requires a numeric stepwise stream at every scored prefix. A trajectory with a nonnumeric value is skipped for that predictor row rather than partially scored or filled with a default. Thus each reported TPS row evaluates complete numeric forecast streams for that predictor; any predictor-specific sample variation is part of the reported transparency convention.

Datasets and run settings. Table 17 summarizes per-dataset configurations. The Tau2 user simulator (Qwen 2.5 7B Instruct) is part of the benchmark interaction environment; it is not the uncertainty-emitting agent scored by TPS. All sampling used seed 42. Tau2-Bench [31] uses 278 attempted base-split episodes, of which 201 completed with a clean terminal outcome; the 201 completed episodes are distinct tasks (50 airline, 113 retail, and 38 telecom). StrategyQA [32] uses the full `wics/strategy-qa` test split. HotpotQA [33] uses a 2000-task sample of the `fullwiki` validation split. WebShop [34] uses the 500-task fixed evaluation subset.

Dataset	Attempted	Completed	Max steps	Seed	Retrieval / action interface
StrategyQA	2,290	2,229	16	42	Local ES (top-k=3, 420 char obs.)
HotpotQA	2,000	1,529	7	42	Local ES (top-k=5, 420 char obs.)
Tau2-Bench	278	201	50	42	Benchmark-native tools; 120-step env. cap
WebShop	500	163	30	42	Text env.; 1,000-product catalog

Table 17: Per-dataset run configuration. *Attempted* counts all launched episodes; *Completed* counts episodes with a clean terminal outcome (Table 7). On Tau2, 201 of 278 attempted episodes completed, each a distinct base-split task; the remaining 77 ended at the step budget (1) or by tool error or environment termination (76). WebShop working sample is 308 (163 completed + 145 admin-censored) after excluding 192 parse-error terminations.

Each of the 201 completed Tau2 episodes is a different task, identified by its `domain::task_id`; the IDs and calibration folds are included in the code repository. Because no task appears twice, resampling episodes in the bootstrap is the same as resampling tasks, and no task appears in both calibration folds. Unless stated otherwise, intervals use 1,000 bootstrap resamples of trajectories. Three analyses resample differently, as described where they appear: the alternative-calibrator comparison holds the calibration folds and fitted maps fixed (Appendix D.1), the rerun-budget analysis resamples both prefixes and reruns (Appendix H.2), and the sample-size analysis draws repeated subsets of the data (Appendix M).

System prompts. The four prompts below are reproduced verbatim. Across all benchmarks, the `<confidence>` field is defined identically: a number in $[0, 1]$ representing the probability that the task will eventually be solved correctly from the current state.

StrategyQA

```
You are a question-answering agent.  You solve yes/no questions by searching a
Wikipedia-backed corpus.
You may use two tools:
1.  Search[query]:  retrieval from the configured Wikipedia backend.  Use it to
discover a relevant page/paragraph and load current passage context.
2.  Lookup[keyword]:  local context scan only (no network).  It scans the
currently loaded passage from the last Search and returns a matching span.
When you have enough information, end with:  Finish[yes] or Finish[no]
At every step, use this exact format:
<think>your reasoning about what to do next</think>
<action>Search[...]  or Lookup[...]  or Finish[yes/no]</action>
<confidence>0.XX</confidence>
<explanation>one sentence explaining your confidence</explanation>
Rules:
- confidence is a number between 0.0 and 1.0 representing the probability that
this task will eventually be solved correctly from the current state.
- Always output all four tags:  think, action, confidence, explanation.
- Use Lookup only after Search has loaded a passage context.
- Prefer Lookup on the current passage before issuing another Search.
- The final answer must be exactly yes or no.
- explanation must not be empty.
- Be concise.
```

HotpotQA

```
You are a question-answering agent.  You solve multi-hop questions by searching
a Wikipedia-backed corpus.
You may use three tools:
1.  Search[query]:  retrieval from the configured Wikipedia backend.  Use it to
discover relevant pages/passages and load context.
2.  Lookup[keyword]:  local context scan only (no network).  It scans the
currently loaded passage and returns a matching span.
3.  Finish[answer]:  terminate with a free-form final answer string.
At every step, use this exact format:
<think>your reasoning about what to do next</think>
<action>Search[...]  or Lookup[...]  or Finish[answer]</action>
<confidence>0.XX</confidence>
<explanation>one sentence explaining your confidence</explanation>
Rules:
- confidence is a number between 0.0 and 1.0 representing the probability that
this task will eventually be solved correctly from the current state.
- Always output all four tags:  think, action, confidence, explanation.
- Use concise targeted search queries; multi-hop questions often require
evidence from multiple pages.
```

```
- Use Lookup after Search to extract precise spans from current context.
- The final answer should be short and direct.
- explanation must not be empty.
- Be concise.
```

Tau2-Bench

```
You are a customer-support agent operating in the tau2-bench domain:
{TAU2_DOMAIN}.
Follow this domain policy exactly:
<domain_policy>
{TAU2_POLICY}
</domain_policy>
You may only use these benchmark-native tools:
<tool_catalog>
{TAU2_TOOL_SPECS}
</tool_catalog>
At every step, output exactly four tags:
<think>brief reasoning</think>
<action>Tool[tool_name, {...valid JSON object...}] or Respond["..."]</action>
<confidence>0.XX</confidence>
<explanation>one sentence</explanation>
Rules:
- Use Tool[...]  only with a listed tool name and a valid JSON object for
arguments.
- Use Respond["..."] to send a plain-text message to the user.
- Output exactly one Action per step.
- confidence must be a number between 0.0 and 1.0.
- explanation must be a non-empty short sentence explaining your confidence.
- Output only these four tags and no extra fields.
- Keep think concise and task-focused.
```

WebShop

```
You are solving a WebShop text-environment task from split:  {WEBSHOP_SPLIT}.
At every step, output exactly four tags:
<think>brief reasoning</think>
<action>Search[query] or Click[target]</action>
<confidence>0.XX</confidence>
<explanation>one non-empty sentence explaining your confidence</explanation>
Rules:
- The action must be exactly one of:  Search[...]  or Click[...].
- Use Search[...]  to submit a search query when the search bar is available.
- Use Click[...]  with a target that exactly matches one of the provided
clickable targets.
- Always emit an <action> tag, even when uncertain.
- Do not emit any other action formats (no Tool[...], Respond[...], Finish[...],
JSON, or plain text actions).
- confidence must be a number between 0.0 and 1.0.
- explanation must be non-empty.
- Do not output extra tags or extra text outside the four required tags.
- Do not repeat search plans or narrate multiple alternatives; choose the single
next action.
```

Reproducibility. Predictor extraction formulas are in Appendix F; parse-error exclusion rules and termination categorization are in Appendix G.

L Existing Assets and Licenses

We used only existing model, benchmark, retrieval, and software assets for evaluation, and we do not introduce a new dataset, model, or benchmark as a contribution. The language-model trajectories were generated with Gemma 4 31B accessed through OpenRouter; the hugging face model card lists the Gemma 4 31B model under the Apache 2.0 license. The StrategyQA benchmark is used under the MIT license reported by its official repository. Tau2-Bench is used under the MIT license reported by its repository. HotpotQA is used under the CC BY-SA 4.0 license reported by the benchmark website. WebShop is used under the MIT license reported by its official repository. The Wikipedia text indexed for local retrieval is used under Wikipedia's content terms, including CC BY-SA 4.0 and GFDL. These assets are used only for evaluation and retrieval in the reported experiments.

M Sample-Size Stability

This appendix asks how many trajectories the main conclusions need. For each dataset we draw 300 random subsets at each smaller sample size and compute bootstrap intervals within each subset (300 resamples), reusing the cached trajectories and reruns. The results describe these datasets only. They are not general minimum sample sizes, which depend on the effect size, how much prefixes differ, the censoring rate, the score, and how precise a conclusion is needed.

Tau2. The TPS improvement from recalibration is positive in every subset, even with only $N = 20$ tasks, and the median improvement stays close to the full-data value of $+5.672$ nats. The full conclusion, that TPS improves while AUROC changes by at most 0.05, holds in 87.3%, 95.3%, 98.3%, and 100% of subsets at $N = 60, 90, 120, 160$. A stricter version requires the whole TPS interval to lie above zero and the whole AUROC interval to lie within $[-0.05, 0.05]$; it holds in 53.0%, 77.7%, and 90.3% of subsets at $N = 90, 120, 160$. The large TPS improvement is therefore visible with few tasks, whereas showing that AUROC barely changes needs more data. The full $N = 201$ dataset is a single data point, so no rate is reported for it.

WebShop. Subsets keep the same mix of completed and budget-stopped trajectories and exclude parse-error stops. The shift from complete-only to censored-aware scoring is positive in every subset. At $N = 40$ its median is $+0.161$ nats, close to the full-sample $+0.159$, and every interval lies above zero. This shows that the result is stable under our stopping classification and the $q_Z \approx 0$ approximation; it says nothing about other stopping mechanisms.

HotpotQA. With $N = 15$ audited prefixes, 99.3% of subsets still show a positive average q_Z and an exact score below the simple score. Under the stricter requirement that both intervals exclude zero, the conclusion holds in 75.3% of subsets at $N = 15$, 96.7% at $N = 25$, and 100% from $N = 40$ onward; at $N = 40$ the median estimates are $\widehat{q}_Z = 0.244$ and a correction of -0.105 nats. These intervals resample prefixes but treat each prefix's q_Z estimate as fixed; uncertainty from the number of reruns is analyzed in Appendix H.2. The two findings are not independent: each prefix's correction equals $\widehat{q}_{Z,i}c_i$, and c_i is negative for nearly every prefix, so a positive $\widehat{q}_Z$ implies a negative correction.

N Reporting Template

Table 18 lists the choices that should be fixed and reported before comparing systems. When the forecasts are probabilities, TPS is the main score, and ranking, selective-prediction, and calibration-error metrics complement it. Reporting several scores or weight schedules is fine as a sensitivity check chosen in advance, but not as a way to pick the most favorable result.

O Extended Practitioner Guide

Section 7 lists the evaluation steps briefly. This appendix explains the reasoning behind each step and then applies all six steps to an example.

O.1 Rationale for each step

1. **What** TPS **scores.** TPS scores a forecast $F_t \in [0, 1]$ made at every step from the information available so far, read as the probability that the task will eventually succeed. Raw

Field	What to report
Score family and parameters	Log, Brier, or beta with (α, β); which score is primary, which are sensitivity checks, and any deployment threshold.
Weight schedule and normalization	Which steps are scored, their weights, how the weights are normalized, and how they are handled on censored prefixes.
Clipping value	The clipping value ε and where it is applied.
Calibration split	The calibration method, how calibration and evaluation data are separated, and how any q_Z estimate is kept apart from the predictor being evaluated.
Stopping classification	How stops were classified (completed, budget, or informative), what was excluded, and how censored prefixes were scored (exact q_Z, labeled $q_Z \approx 0$ approximation, or the $q_Z \in \{0, 1\}$ range).
Resampling unit	What was resampled (task, episode, or stopped prefix), the number of resamples, and whether intervals hold the calibration fits fixed or include rerun uncertainty.

Table 18: Reporting template for the evaluation protocol in Section 7.

signals such as token probabilities or entropies are not yet probabilities; convert them using held-out data that is not used for evaluation (Appendix D). On Tau2-Bench, our conclusions were the same with Platt, isotonic, and beta calibration (Appendix D.1). Summaries that reduce a trajectory to one number, such as Φ_{last} or Φ_{avg}, are not per-step forecasts; evaluate them with scalar metrics instead.

2. **Choosing the score.** With no special decision threshold, use log TPS as the main score and Brier TPS as a bounded check. Because the log score penalizes confident mistakes without limit, report the clipping value. If the system acts when the forecast crosses a known threshold τ, a beta score with its mode at τ, $(\alpha-1)/(\alpha+\beta-2) = \tau$, focuses the evaluation on errors near that decision: for example, Beta(4,8) for a handoff at 0.30 or Beta(5,2) for a verification step at 0.80 (Appendix J.3). Additional scores should be chosen in advance as sensitivity checks, never picked after seeing which looks best.
3. **Choosing the weights.** Any fixed positive weights keep TPS proper; they only decide which steps count most. Uniform weights treat all steps equally, front-loaded weights suit early intervention, and back-loaded weights suit a final check. Weights can also follow the value of an accurate forecast at each step, $w_t \propto v_t$, estimated from domain knowledge or a separate pilot set. Our linear-front schedule represents the early-intervention case, not a universal default. Censored prefixes keep the weights of the full trajectory; they are not rescaled.
4. **Classifying stops.** Label each trajectory as completed, stopped by a fixed step or compute budget, or stopped for a reason that may reveal the outcome, such as a parse failure, a tool error, or a monitor that stops runs that look likely to fail. Report all counts. Only budget stops satisfy Assumptions 1–2, so the others are excluded from censored scoring; rerunning the analysis with them included shows how much this matters (Appendix G.2). Budget stops themselves should not be dropped: on WebShop they were 47% of the working sample and harder than the completed trajectories (Section 6.3).
5. **Scoring censored prefixes.** The exact score needs q_Z, the chance that a stopped prefix would have gone on to succeed. (a) If prefixes can be resumed, rerun them and use the observed success rate as q_Z. In our HotpotQA audit, even two reruns per prefix gave the same dataset-level conclusion in every repetition, although judging an individual prefix needs more (Appendix H.2). (b) If not, q_Z can be predicted by a separate model, as long as it is kept apart from the predictor being evaluated and satisfies Remark C.4. (c) If neither is possible, report $\text{TPS}_{\text{log}}^{\text{cen,simple}}$, labeled as the $q_Z \approx 0$ approximation, together with the scores at $q_Z = 0$ and $q_Z = 1$, which bound the exact score and need no reruns (Remark C.5). Eq. (15) also says how accurate q_Z must be: errors of up to $\varepsilon/|c_z|$ change the log score by at most ε, and if $c_z = 0$ the score does not depend on q_Z. For example, a budget-stopped prefix whose forecasts fall from 0.6 to 0.2 has $c_z \approx -0.40$, so a 0.05-nat tolerance allows q_Z to be off by about 0.13, whereas a prefix that stays at 0.95 allows only about 0.017 (Appendix O.2).

6. **Reporting and interpreting.** Report TPS as the main score and rank and calibration metrics as complements. Give intervals and say what was resampled (task or stopped prefix). Always compare against the base-rate reference: a recalibrated stream can end up barely above an uninformative forecast (Section 6.1), or below it once censored trajectories are included (Appendix I.2, Table 15). Finally, TPS evaluates recorded trajectories under the policy that produced them. It does not measure capability or whether acting on the forecasts would help, and a good score alone does not justify reducing human oversight (Section 8).

O.2 Worked example: a support agent's handoff confidence

A team plans to deploy a customer-support agent that hands the conversation to a human whenever its forecast of eventual success falls below 0.30. They want to compare two confidence streams, the agent's verbal confidence and a lightweight probe on its hidden states, on 500 held-out tasks with a 20-step budget. During evaluation, the agent runs without the handoff rule, so all trajectories come from a single fixed policy. The counts and forecasts below are illustrative; they show how each step of Section 7 is applied.

Step 1: Forecast object. Both streams emit a value at every step, so both are valid TPS inputs once mapped to probabilities. Verbal confidence clusters near 0.9–1.0, so both streams are recalibrated with Platt scaling, fit on a separate labeled set and fixed before the 500 tasks are scored (Appendix D). A trajectory-level summary such as the minimum confidence would instead be evaluated with scalar diagnostics only.

Step 2: Score. Log TPS is the primary score. Because the handoff acts at $\tau = 0.30$, the team also predeclares Beta(4,8), whose mode is $(4-1)/(4+8-2) = 0.30$, as a decision-focused score, and Brier as a bounded check. All three are reported regardless of which stream each favors.

Step 3: Weights. A separate pilot of 40 conversations showed that handoffs after the fifth step rarely rescued the conversation. The team therefore sets $v_t = 2$ for $t \le 5$ and $v_t = 1$ afterwards, with $w_t \propto v_t$ normalized over each trajectory's evaluated horizon. For a 20-step trajectory this gives $w_t = 0.08$ for the first five steps and $w_t = 0.04$ thereafter.

Step 4: Stopping classification. Of the 500 trajectories, 380 complete with an observed outcome, 70 reach the 20-step budget (administrative censoring), and 50 end in tool errors (informative stopping). The 50 tool-error trajectories are excluded from censored scoring, their count is reported, and a sensitivity analysis that treats them as censored shows how much the exclusion matters, without carrying the guarantees of administrative censoring. The 70 budget stops are retained rather than dropped.

Step 5: Censored prefixes. Conversations with live customers cannot be replayed, and no separate model for predicting q_Z is available, so the team reports the simple score labeled as the $q_Z \approx 0$ approximation, together with endpoint intervals. Under the log score, the exact censored score of a prefix is a straight line in q_Z:

$$L_z(q) = L_z(0) + q\, c_z, \qquad L_z(0) = \sum_{t=1}^{z} w_t \log(1 - F_t), \qquad c_z = \sum_{t=1}^{z} w_t \log \frac{F_t}{1 - F_t}.$$

Consider a budget-stopped conversation whose calibrated forecasts were 0.6 for steps 1–5, 0.4 for steps 6–12, and 0.2 for steps 13–20. Then $L_z(0) = -0.581$, $L_z(1) = -0.976$, and $c_z = -0.395$, so the exact score lies in $[-0.976, -0.581]$ whatever q_Z is. Because $c_z < 0$, the $q_Z \approx 0$ approximation sits at the upper end of this interval: it is optimistic for a prefix whose forecasts had already fallen. A tolerance of 0.05 nats permits a q_Z error of up to $0.05/0.395 \approx 0.13$.

Now consider a budget-stopped conversation that forecast 0.95 at every step. Here $c_z = \log(0.95/0.05) \approx 2.94$, the $q_Z \approx 0$ approximation is the pessimistic lower end of $[-3.00, -0.05]$, and the same 0.05-nat tolerance permits a q_Z error of only about 0.017. Overconfident censored prefixes are therefore where estimating q_Z matters most. Averaging the per-prefix lower and upper endpoints over the 70 censored trajectories gives the censored-subset interval. Combining those bounds with the fixed scores of the 380 completed trajectories gives the full working-sample interval over 450 trajectories.

Step 6: Reporting and interpretation. Table 19 shows the completed reporting template. Both streams are compared against the base-rate reference, with task-level bootstrap intervals. If the probe attains higher TPS, the conclusion is that its probabilities are more truthful under the current policy. Whether switching the handoff rule to the probe would resolve more conversations is a closed-loop question that requires an online experiment.

Field	Example entry
Score family and parameters	Log (primary); Beta(4,8) for the 0.30 handoff gate and Brier as predeclared sensitivity scores.
Weight schedule and normalization	$w_t \propto v_t$ with $v_t = 2$ for $t \leq 5$ and 1 otherwise, normalized over each evaluated horizon T_i; budget-stopped prefixes use the 20-step horizon.
Clipping value	$\varepsilon = 10^{-6}$, applied to all forecasts before any log score.
Calibration split	Platt scaling on a separate labeled calibration set, with no evaluation tasks used for fitting; no q_Z estimate.
Stopping classification	380 complete; 70 administrative (20-step budget); 50 informative (tool errors), excluded and reported. Censored prefixes scored with the labeled $q_Z \approx 0$ approximation plus endpoint intervals.
Resampling unit	Task; 1,000 bootstrap replicates; intervals conditional on the fitted calibration maps.

Table 19: Completed reporting template (Table 18) for the worked example.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's main claims, and the theory and experimental sections scope those claims to proper scoring of prefix-conditioned success probabilities.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 8 discusses the main limitations, including administrative censoring assumptions, binary-outcome scope, and the single-model, single-harness empirical setting.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main theoretical assumptions and results are stated in Sections 3–5; full proofs are provided in Appendices A, B, C, and J.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 6 describes the benchmarks, model, harness, predictors, scoring choices, and censoring protocol; Appendices D, G, I, and K provide additional reproducibility details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and data are provided at `https://github.com/R-Suresh07/proper-scoring-agentic-uq`. The repository includes scoring and calibration implementations, experiment scripts, numerical analysis inputs, benchmark configurations, prompts, domain-qualified Tau2 task IDs and calibration folds, and setup and reproduction instructions. External benchmark environments and source datasets are obtained separately using the documented setup instructions; raw interaction logs are not included. The README distinguishes bundled reference results from reruns and documents numerical rank-statistic differences.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 specifies the datasets, splits, predictor streams, score families, and primary settings; Appendices D and K provide calibration and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Primary experiments report 95% intervals from 1000 trajectory-level bootstrap resamples. Appendices D.1, H.2, and M specify the conditioning and resampling procedures for calibrator, continuation-budget, and sample-size analyses.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix K reports the hosted inference setup, model, decoding hyperparameters, benchmark-specific run settings, fixed step budgets, seeds, and retrieval/tooling setup. Trajectories were generated through OpenRouter. The paper does not report complete per-experiment memory and runtime measurements or total compute usage.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work evaluates existing benchmark trajectories and scoring rules and does not require deviations from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Section 8 discusses the risk of mistaking probability quality for capability or using favorable scores to reduce human oversight, together with held-out and distribution-shift evaluation. A broader account of positive and negative societal impacts remains outside the scope of this evaluator study.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk pretrained model, image generator, scraped dataset, or other asset requiring controlled release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix L credits the existing model, benchmark, retrieval, and software assets used in the evaluation and states their license or source terms in prose, including Apache 2.0 for Gemma 4 31B, MIT for StrategyQA, Tau2-Bench, and WebShop, CC BY-SA 4.0 for HotpotQA, and Wikipedia's CC BY-SA 4.0/GFDL content terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released evaluation code and analysis artifacts are accompanied by a README, setup instructions, experiment commands, configuration files, and an Apache 2.0 license for the project code at `https://github.com/R-Suresh07/proper-scoring-agentic-uq`. The documentation describes the included numerical inputs and the external benchmark dependencies. We do not introduce a new source dataset, model, or benchmark.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subjects research, so IRB or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Section 6 declares the evaluated language model and ReAct harness because LLM-agent trajectories are central to the empirical evaluation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.