WorldFork: Auditable Branching Rollouts for LLM Forecasting

Hanson Wen [1] Shing Cheung James Gui [2]

Abstract

Forecasting agents should produce not only calibrated probabilities, but also inspectable records of the assumptions and evidence that support those probabilities. We introduce WorldFork, an auditable branching-rollout protocol for LLM forecasting that turns a public event card into a multiverse of branching timelines with actors, constraints, candidate endpoints, branch lineage, path mass, endpoint ledgers, and report provenance. On 24 masked retrospective resolved-event cards, unconditional branching reduces WorldFork Brier score from 0.282 to 0.214, improves log score from 0.725 to 0.581, and yields lower per-card Brier loss than no-branch rollouts on 17 of 24 cases under an exact zero-difference comparison; a fixed 50/50 blend with a direct JSON forecast reaches Brier 0.205. These results are pilot evidence rather than a confirmatory benchmark: retrospective masking only partially controls leakage, the sign test is suggestive but not significant ($p = 0.064$), and a card-bootstrap confidence interval includes zero. A post-hoc error analysis suggests a bias–variance pattern in which branching preserves otherwise-missed endpoint mass, but can dilute no-branch forecasts that were already decisive and correct. The main contribution is therefore a forecast object and audit protocol that make branching assumptions, unresolved mass, and mechanism claims inspectable for future locked evaluations.

[1]University of California, Berkeley [2]University of Southern California. Correspondence to: Hanson Wen <hansonwen@berkeley.edu>, Shing Cheung James Gui <jamesgui@usc.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Forecasting is an attractive testbed for general-purpose AI because it requires temporal reasoning, calibration, and uncertainty management in dynamic real-world settings.[1] Yet a useful forecast is not just a scalar probability. When forecasts inform decisions, reviewers also need to inspect which evidence was used, which counterfactual assumptions were considered, how much probability mass remains unresolved, and whether the narrative explanation is grounded in the trace.

WorldFork represents a forecast as a branching trace rather than as a single answer. A public event card is converted into actors and cohorts, constraints, binary endpoints, graph state, social and institutional variables, alternate timelines, path masses, endpoint ledgers, and report provenance. Generated futures are treated as structured hypotheses, not observations. This representation makes the assumptions behind a probability auditable while preserving compatibility with proper scoring rules.

This paper contributes an auditable branching-rollout protocol, a benchmark workflow that separates direct scalar forecasts from rollout-based extraction, and a 24-card retrospective pilot comparing branching and no-branch WorldFork rollouts. The empirical result is encouraging but deliberately modest: unconditional branching improves aggregate descriptive scores in the available artifacts, but the sample is small, retrospective leakage remains possible, and exploratory blend comparisons should be rerun on locked future cards. The per-card pattern further suggests that branching should be studied as an adaptive intervention rather than as a default replacement for a single settled rollout.

2. WorldFork Protocol

WorldFork operationalizes a forecast as a structured multiverse rollout with explicit endpoint semantics. A multiverse is a tree of timelines rooted in the same public scenario: each timeline records its own state,

[1]https://github.com/Hilo-Hilo/worldfork-icml.

evidence, endpoint ledger, and path mass, while child timelines inherit the parent history up to the fork point and then evolve independently. Appendix A gives a compact end-to-end description of this process.

Initialization. A public scenario card is converted into typed actors and cohorts, scenario constraints, candidate endpoints, graph layers, and an initial social state. Resolved-card scoring uses explicit binary yes/no endpoints. Auxiliary mechanism endpoints may explain how an outcome could occur, but they are not scored as the primary binary outcome and can introduce leakage if they encode mechanisms that are only evident after resolution.

Ticks and branches. A no-branch policy follows one timeline. A branching policy allows alternate continuations inside the multiverse, bounded by branch depth, active timelines, branches per tick, and branch-score thresholds. Path mass records the probability assigned to each surviving branch. Branches are useful only if their endpoint state remains well formed: parent settlements should be inherited unless a child forks before decisive evidence appears, and open-ended simulation closure must not be confused with deadline settlement.

Information locality. Direct LLM baselines receive the full public card at once. WorldFork decomposes the card into local actor/cohort views and advances those views through trace updates. This design may reduce one look-ahead channel, but it does not prove lower leakage overall because the underlying models may still carry latent knowledge and auxiliary endpoints can introduce resolution-adjacent clues.

Ledgers and reports. Endpoint ledgers track whether evidence supports, contradicts, or leaves open each candidate outcome. For scoring, WorldFork converts path-level endpoint settlements into p_{yes} and reports unresolved mass separately. Brier is $(p_{\text{yes}} - y)^2$; log score uses the same binary probability clipped to $[0.01, 0.99]$.

WorldFork can attach social-interaction traces to public attention, propagation, and group-response dynamics. OASIS by CAMEL-AI is related background for LLM/rule-based social simulation; WorldFork's distinction is forecast-specific audit semantics: branch lineage, endpoint ledgers, path mass, and deadline-aware scoring (Yang et al., 2024).

3. Benchmark Design and Scoring

The benchmark contains 108 public cards: 72 synthetic stress prompts and 36 forecasting cards. The forecasting set contains 24 retrospective resolved forecast cards, 8 longform dossiers, and 4 adversarial/calibration cards; only the 24 resolved cards are scored with proper scoring rules. Public inputs mask salient entity names and withhold private resolutions until predictions are frozen.

Leakage controls. Retrospective evaluation creates forward-looking bias risk because resolved 2024–2026 events may appear in training data or retrieval corpora. Masking reduces exact-name matching but does not eliminate latent memorization or pattern recognition from public context. WorldFork decomposition limits some full-card exposure, but that design choice is not sufficient to justify a lower-leakage claim without future locked cards and pre-registered extraction rules.

Baselines. We compare uniform binary prediction, direct JSON API calls, an ai-prophet package baseline (AI Prophet Team, 2026), no-branch WorldFork, branching WorldFork, and a fixed blend. The WorldFork rows use a mixed model policy: DeepSeek v4 Flash for cohort agents and GPT-5.4-Codex/GPT-5.4 for initialization, global review, and other non-cohort roles. Several baselines, metrics, extraction rules, and blend analyses were explored, so Table 1 is descriptive pilot evidence rather than a family of confirmatory hypothesis tests.

Scoring separates probabilistic accuracy from audit diagnostics. For each resolved card, Brier is the mean of $(p_i - y_i)^2$ and log score is binary negative log likelihood after clipping probabilities to $[0.01, 0.99]$. The fixed blend uses $p_i^{\text{blend}} = \frac{1}{2}p_i^{\text{direct}} + \frac{1}{2}p_i^{\text{WF}}$. Appendix J reports the exact per-card probabilities used for consistency checks.

For WorldFork, unresolved mass contributes neutrally when it cannot be settled as yes or no and remains an audit diagnostic rather than a third scored class. This distinction matters because high unresolved mass can make a rollout look cautious while hiding weak endpoint settlement. We therefore report unresolved mass alongside proper scores and do not treat it as an outcome category. The primary paired statistic is

$$\Delta_{\text{Brier}} = \text{Brier}_{\text{no-branch}} - \text{Brier}_{\text{branching}}, \qquad (1)$$

so positive values favor branching. We count a card as branching lower-loss when branching has lower per-card Brier loss than no-branch under an exact zero-difference comparison. This descriptive count is sensitive to numerical margins; practical-tie thresholds should be pre-registered in future runs. For the current 17/24 lower-loss count, the exact two-sided sign test gives $p = 0.064$. A nonparametric card bootstrap over paired Brier differences gives a 95% percentile in-

Table 1. Resolved-card scores computed from the per-card probabilities in Appendix J. WF mixed denotes DeepSeek v4 Flash for cohort agents and GPT-5.4-Codex/GPT-5.4 for initialization, global review, and other non-cohort roles. Exploratory rows are not confirmatory comparisons.

Condition	Model	n	Brier	Log	Role
Uniform binary	fixed	24	0.250	0.693	reference
Direct JSON	GPT-5.4	24	0.236	0.677	baseline
Direct JSON	DeepSeek v4 Flash	24	0.260	0.741	exploratory
AI Prophet package	GPT-5.4	24	0.245	0.715	exploratory
No-branch WF	WF mixed	24	0.282	0.725	pilot
Branching WF	WF mixed	24	0.214	0.581	pilot
Fixed 50/50 blend	GPT-5.4+WF	24	0.205	0.527	exploratory

Table 2. Primary matched branch/no-branch result. The bootstrap interval is over cards and uses the per-card Brier differences in Appendix J.

Statistic	Value
Matched pairs	24
No-branch Brier	0.282
Branching Brier	0.214
Δ_{Brier}	0.068
Bootstrap 95% CI	$[-0.065, 0.189]$
Branching lower-loss cards, exact	17 / 24
Sign-test p-value, two-sided	0.064
Branching log score	0.581
Average unresolved mass	0.27

terval of $[-0.065, 0.189]$ for Δ_{Brier}, so the pilot does not establish a statistically significant effect.

4. Results

All 108 public cards have live initialization evidence: 108/108 successful initializations, mean actor count 8.82, mean graph edge count 34.21, mean rubric score 3.884 on a 0–4 scale, and no critical failures. This structural proxy tests runtime coverage rather than forecast accuracy.

Paired branch/no-branch effect. Branching improves mean Brier from 0.282 to 0.214, giving $\Delta_{\text{Brier}} = 0.068$ and a 24.1% relative reduction. It has lower per-card Brier loss on 17 of 24 cards under an exact zero-difference comparison. The exact two-sided sign-test p-value is 0.064, and the paired card-bootstrap 95% interval is $[-0.065, 0.189]$. The result is therefore directionally encouraging but remains a pilot finding.

Post-hoc branch-loss anatomy. The seven no-branch lower-loss cards are not homogeneous. Two are numerical near-ties: cases 022 and 024 differ by only 1.68×10^{-5} and 9.22×10^{-5} in Brier loss. Four substantive losses share a common pattern: no-branch was already confident and correct, while branching diluted the endpoint probability toward uncertainty. In cases 006, 012, and 016, the true label is yes and no-branch assigns $p_{\text{yes}} = 0.9842$, 0.9233, and 1.0000, while branching assigns 0.4872, 0.2622, and 0.0680. In case 020, the true label is no and no-branch assigns $p_{\text{yes}} = 0.0972$, while branching assigns 0.4496. Case 015 is a both-wrong comparison in which no-branch is less wrong, but neither method assigns majority mass to the true outcome. This pattern suggests that branching is best viewed as an uncertainty-expansion step: it can preserve missed endpoint mass, but it can also inject variance when the initial rollout has already settled the endpoint correctly. Appendix E gives the per-card diagnostic table.

Complementarity with direct forecasts. Branching has lower reported Brier than direct JSON GPT-5.4, 0.214 versus 0.236. A fixed 50/50 blend reports Brier 0.205. This blend result is exploratory: multiple baselines and extraction choices were inspected, so the comparison should be rerun on locked future cards before it is treated as evidence of deployable forecasting skill.

Unresolved mass. Branching has average unresolved mass 0.27. This quantity is useful audit information, but it can hide weak endpoint settlement and pull a probability toward 0.5. The branch-loss anatomy in Appendix E is one reason to report unresolved mass alongside Brier rather than as a scored endpoint.

5. Mechanism Auditability

WorldFork's value is not limited to the final scalar probability. Forecast reports often contain mechanism claims such as public anger rising, trust collapsing, coalitions forming, or regulators becoming more likely to intervene. Such claims can be persuasive even when unsupported. WorldFork treats them as audit objects: a claim should be traceable to a case, branch, tick, actor or cohort, source text, extracted state, confidence, and evidence reference. When that support is missing, the report should mark the claim as speculative.

This layer is not scored with Brier; its target is grounding. A reviewer can sample final-report claims and label them supported, partially supported, unsupported, or not checkable. If a report claims that backlash increased before a protest branch, the trace should contain the corresponding posts, comments, actor summaries, or social-interaction events. If a report says settlement became likely because trust increased, graph or actor-state logs should show that shift before endpoint mass changes.

The audit surface also helps diagnose failures. A direct scalar call may output an accurate probability and a plausible explanation without preserving enough evidence to audit that explanation. WorldFork can be wrong in probability while still exposing why: missing deadline evidence, inherited ledger state, unsupported mechanism claims, auxiliary endpoint leakage, or unresolved official text. This property is especially relevant when forecasting agents operate in open-ended real-world settings where downstream users need to know which assumptions carry the forecast.

Audit scorecard. We propose reporting an audit scorecard next to conventional forecast scores. Endpoint settlement validity checks whether each yes/no probability came from path-level settlement rather than mixed ledger rows. Path-mass conservation checks whether yes, no, and unclear masses sum to one across active branches. Report grounding samples final-report claims and asks whether each is supported by tick events, source packets, endpoint-ledger evidence, or actor/cohort summaries. Social-mechanism grounding asks whether claims about trust, backlash, legitimacy, mobilization, or emotion are visible in the trace before endpoint mass changes. Finally, leakage exposure records whether a prediction came from a full-card direct call, a package baseline, or a local-cohort rollout.

6. Discussion and Limitations

The central lesson is that a branching trace is not automatically a better forecast distribution. Proper scoring requires a clean probability over scored endpoints, not semantically mixed ledger rows. WorldFork's current result is encouraging but modest: it improves over no-branch in the available per-card table, does not reach nominal sign-test significance, has a bootstrap interval that includes zero, and yields an exploratory blend improvement that is not a corrected multi-comparison result. The branch-loss cases point to a bias–variance tradeoff: branching can correct rollouts that collapse too early onto the wrong endpoint, but it can also dilute already-correct high-confidence forecasts when speculative or unresolved branches retain too much mass.

The design requirements are concrete. Binary deadline settlement must be separate from open-ended simulation closure; scoring must be path-level; auxiliary mechanisms should not be scored as primary endpoints; child branches should inherit already-settled parent endpoints unless they fork before decisive evidence; and event summaries must preserve the official text needed to settle the card. The branch-loss pattern also motivates adaptive branching: a future locked protocol can run a no-branch endpoint pass first and fork only when endpoint confidence or ledger support is insufficient. The present pilot should be treated as the development set that motivates this rule, not as a confirmatory test of it.

Retrospective masking still matters, but it cannot eliminate leakage risk. The benchmark is a controlled stress test for forecast extraction, not a clean live forecasting benchmark. Future locked evaluations should release public cards before resolution, freeze predictions, pre-register extraction rules, pre-register any practical-tie threshold for paired lower-loss counts, report exact per-card predictions, and include paired confidence intervals plus a multiplicity plan.

Multiple-comparison risk is also central. The paper reports several baselines, two proper scoring rules, WorldFork extraction variants, and a fixed blend. The blend is useful because it suggests branch mass may contain a signal different from direct scalar probability, but it is not a multiplicity-corrected discovery. The next evaluation should pre-register a primary endpoint, baseline set, extraction rule, blend policy if any, and significance procedure before labels are opened.

The model split should also be locked more explicitly. Direct JSON rows use single full-card model calls. WorldFork rows use DeepSeek v4 Flash for cohort agents and GPT-5.4-Codex/GPT-5.4 for initialization, global review, and other non-cohort roles. A future benchmark should report provider, requested model, resolved model snapshot, prompt version, and response identifier for every scored call so model routing cannot become an untracked degree of freedom.

7. Conclusion

WorldFork is best framed as an auditable forecasting substrate with an encouraging paired pilot, not as a finished replacement for direct LLM forecasting. On 24 matched resolved cards, branching reduces WorldFork Brier from 0.282 to 0.214 and achieves log score 0.581. The observed pattern supports the hypothesis that branching rollouts may add value when endpoint semantics are explicit, leakage exposure is audited rather than assumed, and traces remain inspectable. The pilot also shows that branching should not be unconditional: the branch-loss cases concentrate in settings where no-branch forecasts were already confident and correct. Future locked evaluations should therefore test adaptive branching policies that fork under uncertainty while preserving decisive endpoint settle-

ments.

Impact Statement

This paper studies auditable forecasting agents. More transparent forecasting systems could help users inspect uncertainty, evidence, and assumptions before making decisions. The same systems could cause harm if retrospective pilot results are over-interpreted, if simulated branches are treated as evidence, or if scalar probabilities are used without domain expertise in high-stakes settings. We therefore emphasize leakage controls, disclosure of unresolved mass, grounding of mechanism claims, and human review. The work is intended as an evaluation protocol and research substrate, not as a deployed decision system.

References

Brier, G. W. Verification of forecasts expressed in terms of probability. Monthly Weather Review, 78(1):1–3, 1950.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477):359–378, 2007.

Tetlock, P. E. and Gardner, D. Superforecasting: The Art and Science of Prediction. Crown, 2015.

AI Prophet Team. ai-prophet: AI Prophet ecosystem CLI and Prophet Arena trade benchmark runner. PyPI package, 2026.

Yang, Z., Zhang, Z., Zheng, Z., Jiang, Y., Gan, Z., Wang, Z., Ling, Z., Chen, J., Ma, M., Dong, B., Gupta, P., Hu, S., Yin, Z., Li, G., Jia, X., Wang, L., Ghanem, B., Lu, H., Lu, C., Ouyang, W., Qiao, Y., Torr, P., and Shao, J. OASIS: Open Agent Social Interaction Simulations with One Million Agents. arXiv:2411.11581, 2024.

A. How WorldFork Works

WorldFork treats forecasting as constructing and scoring a tree of possible futures rather than asking a model for a single probability in one pass. The input is a public scenario card with an as-of date, a forecast deadline, and one or more candidate endpoints. Initialization creates the root scenario state, which we call the Big Bang: it stores the scenario text, forecast contract, actors and cohorts, constraints, graph and social variables, seed events, branch hypotheses, and an initial endpoint ledger. The first timeline in the multiverse starts from this root state.

The rollout then advances each active timeline through discrete ticks. During a tick, actor and cohort agents receive local state, propose decisions or events, and update social or institutional variables. A governance/review step audits the provisional tick bundle, applies endpoint-ledger updates, executes structured tool calls, and decides whether the timeline should continue, terminate, or fork. A fork creates a child timeline that inherits parent ticks, future queued events, actor/cohort state, graph context, and relevant prompt influences up to the fork point. After the fork, the child has its own executable state, so two timelines can preserve different hypotheses about the same unresolved event.

Branching is constrained by a policy rather than left open-ended. The policy caps branch depth, the number of active timelines, branches per tick, and branch-score thresholds. Each retained timeline carries path mass, which records how much probability remains attached to that timeline after splits and termination. At scoring time, endpoint ledgers answer whether each timeline supports yes, no, or unresolved for the binary forecast contract, while path mass determines how much of the retained multiverse supports each status. Unresolved mass is reported as an audit quantity and is handled neutrally in the binary probability extraction. Final reports compare terminal timelines, summarize the outcome distribution, and attach evidence so reviewers can inspect which branch assumptions and mechanism claims carried the forecast.

B. Reported Score Values

This appendix repeats the rounded score values used in Table 1. Values are computed from the per-card probabilities in Appendix J; exploratory rows are included for transparency rather than confirmatory testing.

Table 3. Rounded score values reported in Table 1. All scored rows use the same 24 resolved cards. Exploratory rows are reported for transparency but are not confirmatory tests.

Condition	Model	n	Brier	Log	Role
Uniform binary	fixed	24	0.250	0.693	reference
Direct JSON	GPT-5.4	24	0.236	0.677	baseline
Direct JSON	DeepSeek v4 Flash	24	0.260	0.741	exploratory
AI Prophet package	GPT-5.4	24	0.245	0.715	exploratory
No-branch WF	WF mixed	24	0.282	0.725	pilot
Branching WF	WF mixed	24	0.214	0.581	pilot
Fixed 50/50 blend	GPT-5.4+WF	24	0.205	0.527	exploratory

Table 4. Resolved-card construction summary, cases 001–012. Source/risk abbreviates the private resolution source class and leakage-risk label.

ID	y	As-of	Deadline	Domain; source/risk
001	yes	2025-09-15	2025-10-10	institutions/awards; Nobel/med
002	yes	2025-09-15	2025-10-08	science/awards; Nobel/med
003	yes	2025-11-15	2025-12-10	macro policy; Fed/low
004	no	2026-01-10	2026-01-28	macro policy; Fed/low
005	no	2026-01-23	2026-03-15	culture/awards; Academy/med
006	yes	2025-11-10	2026-02-01	culture/awards; Grammy/med
007	no	2025-10-24	2025-11-01	sports; MLB/low
008	no	2026-02-06	2026-02-08	sports; team/league/low
009	yes	2026-01-28	2026-02-01	sports; AusOpen/low
010	no	2026-01-29	2026-01-31	sports; AusOpen/low
011	yes	2025-10-01	2025-11-04	election; news/med
012	yes	2025-10-01	2025-11-04	election; AP/NPR/med

C. Resolved-card Construction

Each resolved card was created from a real-world event between 2024 and 2026. We first wrote a public model-facing card containing only pre-resolution information, an as-of date, a deadline, and binary candidate endpoints. We then wrote a private evaluation object with the true resolution, resolution date, masked-entity map, and resolution source URLs. Entity names were masked before model input to reduce exact memorization and answer lookup. The private resolution file was not shown to forecasting systems until predictions were frozen.

D. Statistical Checks

This section documents the scoring and statistical rules used for the rounded values in the main paper. All proper-score rows use the same 24 resolved forecast cards. Cards are not reweighted by topic, source type, or confidence. The added 36-card file is not a 36-case proper-scoring set: it contains 24 resolved forecast cards, 8 longform dossiers, and 4 adversarial/calibration cards. The latter 12 do not have binary yes/no resolutions and should not be added to

Table 5. Resolved-card construction summary, cases 013–024. Public masked IDs and exact resolution URLs are stored in the private resolution-source artifact.

ID	y	As-of	Deadline	Domain; source/risk
013	no	2025-10-01	2025-11-04	election; AP/NPR/med
014	no	2025-04-01	2025-04-28	election; parliament/med
015	yes	2025-01-17	2025-06-30	product launch; company/med
016	yes	2025-09-10	2025-09-30	product launch; company/low
017	yes	2025-07-01	2025-10-14	tech policy; vendor/low
018	yes	2025-04-01	2025-04-30	regulation; regulator/med
019	yes	2025-07-25	2025-08-31	merger; company/low
020	no	2024-12-01	2024-12-31	antitrust merger; FTC/company/high
021	no	2026-02-20	2026-03-31	space launch; agency/low
022	no	2025-12-15	2026-01-31	climate record; NOAA/NCEI/low
023	no	2025-11-10	2025-11-22	climate negotiation; UN/low
024	no	2025-08-15	2025-09-30	antitrust remedy; court/news/low

the Brier/log-score denominator.

For a method m, the reported Brier score is

$$\text{Brier}(m) = \frac{1}{24}\sum_{i=1}^{24}(p_{m,i} - y_i)^2, \tag{2}$$

where $p_{m,i}$ is the frozen yes probability for card i and $y_i = 1$ for a private yes resolution and $y_i = 0$ for a private no resolution. The reported log score is

$$\text{Log}(m) = \frac{1}{24}\sum_{i=1}^{24}\big[-y_i\log(\tilde{p}_{m,i}) - (1-y_i)\log(1-\tilde{p}_{m,i})\big], \tag{3}$$

where $\tilde{p}_{m,i} = \min(0.99, \max(0.01, p_{m,i}))$ is the clipped probability used to avoid infinite loss at exactly 0 or 1. The per-card table in Appendix J recomputes the rounded main values: no-branch WorldFork Brier/log 0.282/0.725, branching WorldFork 0.214/0.581, and fixed 50/50 blend 0.205/0.527.

WorldFork prediction extraction first converts branch/path evidence into the binary probability consumed by these formulas. For candidate-status mass extraction, yes endpoint rows contribute realized mass to yes and eliminated mass to no; no endpoint rows contribute realized mass to no and eliminated mass to yes. If unresolved candidate mass remains, the extracted probability is

$$p_{\text{yes}} = \frac{M_{\text{yes}} + 0.5M_{\text{unresolved}}}{M_{\text{yes}} + M_{\text{no}} + M_{\text{unresolved}}}, \tag{4}$$

with $p_{\text{yes}} = 0.5$ when no binary endpoint mass is available. For path-ledger extraction, each included path contributes its normalized weight times that path's extracted p_{yes}; uncertain paths use the same neutral 0.5 default unless a deadline-forced value is explicitly supplied. The resulting scalar p_{yes} is what Brier and log score use. The unresolved mass is retained in the reported predictions and tables as an audit quantity, not as an additional outcome class.

The paired branch/no-branch lower-loss count is computed per card from Brier losses: branching is lower-loss on card i if $(p_{\text{branch},i} - y_i)^2 < (p_{\text{no-branch},i} - y_i)^2$. The primary descriptive count in this pilot uses an exact zero-difference comparison because no practical-equivalence threshold was pre-registered. In future runs, ties and practical ties should be excluded from the sign-test denominator or reported separately before computing the binomial test. In the current 24-card pilot there are 17 branching lower-loss cards out of 24 matched cards, giving an exact-comparison lower-loss rate of $17/24 = 70.8\%$.

The exact sign-test calculation treats the number of branching lower-loss cards L as Binomial$(n = 24, q = 0.5)$ under the null hypothesis that branching and no-branch are equally likely to have lower loss on a card. The two-sided p-value for the observed favorable count is

$$p_{\text{sign}} = 2\sum_{k=17}^{24}\binom{24}{k}2^{-24} = 0.063914656639, \tag{5}$$

reported as 0.064 after rounding. This is suggestive but not statistically significant at the conventional 0.05 level, so the result should be treated as an initial pilot evaluation rather than a final statistical claim.

Cases 022 and 024 illustrate why future runs should pre-register a practical-tie rule. Their no-branch advantages are 1.68×10^{-5} and 9.22×10^{-5} in Brier loss, respectively. Under a post-hoc practical-equivalence threshold of $\epsilon = 0.001$ in per-card Brier-loss difference, these two cards would be counted as ties, leaving 17 branching lower-loss cards among 22 non-tied comparisons and giving a two-sided sign-test value of $p = 0.017$. This sensitivity is diagnostic only and is not used as the primary statistical result.

Table 6. Diagnostic sensitivity of the paired lower-loss count to practical-tie thresholds. Thresholds were not pre-registered, so the exact zero-difference row remains the primary descriptive count.

ϵ	Branch	No-branch	Ties	p_{sign}
0	17	7	0	0.064
0.0001	17	5	2	0.017
0.001	17	5	2	0.017
0.01	15	5	4	0.041
0.05	12	5	7	0.143

A paired nonparametric bootstrap was computed by resampling the 24 matched cards with replacement and recomputing the mean per-card Brier difference Δ_{Brier}

for each draw. Using the per-card probabilities in Appendix J, the observed mean difference is 0.0679 and the 95% percentile interval is $[-0.065, 0.189]$. The interval includes zero, reinforcing that the pilot is not a confirmatory result.

Multiplicity and analysis degrees of freedom. The benchmark development process considered several baselines, two proper scoring rules, multiple WorldFork extraction variants, and at least one blend. The paper therefore does not treat the smallest observed Brier value or the fixed-blend comparison as a multiplicity-corrected discovery. A locked follow-up should pre-register the primary endpoint, extraction rule, baseline set, blend policy if any, and statistical test before seeing private labels.

Leakage and endpoint-audit scope. Retrospective masking lowers exact-name exposure but does not eliminate latent model memory or resolution-pattern recognition. WorldFork decomposition should be evaluated as a process design rather than a proven lower-leakage intervention. Auxiliary endpoints are useful for mechanism audit, but they can leak information if they name mechanisms that are only obvious after resolution. Future runs should audit auxiliary endpoints for resolution-adjacent wording before scoring and preserve the rejected endpoint list.

E. Post-hoc Branch-Loss Anatomy

This appendix classifies the seven cards on which no-branch has lower Brier loss under the exact comparison. The categories are descriptive and were assigned after inspecting the frozen probabilities in Appendix J; they should be treated as diagnostics rather than pre-registered subgroup results.

Table 7. No-branch lower-loss cards. Here $\Delta_i = (p_{\mathrm{NoWF},i} - y_i)^2 - (p_{\mathrm{BrWF},i} - y_i)^2$, so negative values favor no-branch.

Case	y	NoWF	BrWF	Δ_i	Interpretation
006	1	0.9842	0.4872	−0.263	Confident correct forecast diluted by branching
012	1	0.9233	0.2622	−0.538	Confident correct forecast diluted by branching
015	1	0.2530	0.1803	−0.114	Both methods assign minority mass to the true outcome; no-branch is less wrong
016	1	1.0000	0.0680	−0.869	Confident correct forecast diluted by branching
020	0	0.0972	0.4496	−0.193	Confident correct forecast diluted by branching
022	0	0.0079	0.0089	-1.68×10^{-5}	Numerical near-tie
024	0	0.0000	0.0096	-9.22×10^{-5}	Numerical near-tie

The substantive pattern is concentrated in cases 006, 012, 016, and 020. In those cards, the no-branch rollout is already on the correct side with high confidence, and branching pulls the final probability back toward uncertainty. Case 015 is a different pattern: both methods are on the wrong side of the binary outcome, with no-branch merely less wrong. Cases 022 and 024 are better treated as practical ties unless a future protocol pre-registers an exact zero-difference rule. This motivates an adaptive follow-up in which branching is triggered by uncertainty or weak ledger support rather than applied to every card.

F. Baseline Prompts and Run Settings

The direct baselines used one full public-card call per resolved case, no retrieval, temperature 0.2, max tokens 4096, and JSON-schema output validation. GPT-5.4 calls used the openai-codex provider route; DeepSeek calls reused the same prompt families through OpenRouter. The shared system prompt was:

> You are a calibrated forecasting assistant.
> You are evaluating a resolved-but-hidden event, but you do not know the resolution.
> Use only the public card text provided by the user.
> Do not use web search.
> Do not infer from real-world memory if the card is entity-masked. Return valid JSON only.

The direct JSON user payload contained the public card, asked for one calibrated yes/no forecast and short rationale, and required an identifier, condition label, yes/no probabilities, confidence, main drivers, main uncertainties, and a leakage warning. The structured variant used the same public card but separated evidence for and against the event, base-rate analogies from the card only, key uncertainties, a calibration note, yes/no probabilities, and a leakage warning. Both prompt families enforced: probabilities sum to one, do not mention or imply the true resolution, and do not use web or private evaluation data.

For WorldFork short rollouts, the no-branch policy disabled branching by allowing one active timeline and using a high branch threshold of 0.999. The branching policy allowed depth 2, up to four active timelines, up to two branches per tick, a branch threshold of 0.75, at least two runway ticks, and candidate-endpoint branches only. Deadline-aware runs used the public as-of date and forecast deadline to set the tick duration and horizon.

G. Model and Run Identifiers

Exact model IDs are not uniformly locked across all rows. Table 9 records the identifiers available from the reported runs and the remaining reproducibility

Table 8. Prompt and run-setting summary for the reported forecast rows. WorldFork mixed route denotes DeepSeek v4 Flash for cohort work and GPT-5.4-Codex/GPT-5.4 for initialization, global review, endpoint-ledger, and report work.

Method	Provider / route	Model label	Temp.	Retrieval	Calls	Output schema
Direct JSON GPT-5.4	OpenAI Codex	GPT-5.4	0.2	no	24	direct yes/no JSON plus rationale
Structured direct GPT-5.4	OpenAI Codex	GPT-5.4	0.2	no	24	structured evidence plus yes probability
Structured direct DeepSeek	OpenRouter	DeepSeek v4 Flash	0.2	no	24	same structured schema
ai-prophet GPT-5.4	OpenAI	GPT-5.4 (2026-03-05)	default	no	24	FORECAST-stage JSON
WorldFork no-branch	WF mixed route	mixed	route	no	24 rollouts	endpoint-ledger/path-mass extraction
WorldFork branching	WF mixed route	mixed	route	no	24 rollouts	endpoint-ledger/path-mass extraction

gap. Future locked evaluations should report provider, requested model, resolved model snapshot, route policy, prompt version, and response identifier for every scored call.

H. AI Prophet Baseline Details

The ai-prophet baseline used the package's FORECAST-stage style: an expert forecaster prompt returns a JSON yes probability and a short rationale. For this static benchmark, SEARCH and ACTION were disabled; each public masked resolved card was sent once to OpenAI GPT-5.4, and only the returned yes probability was scored. The run produced 24 predictions, averaged 3784.5 ms per call at concurrency four, and used 16,669 total tokens.

The artifact manifest records provider openai, requested model gpt-5.4, resolved model gpt-5.4-2026-03-05, 13,454 input tokens, 3,215 output tokens, and 16,669 total tokens. The package version and exact shell command were not captured in the manifest, so they remain reproducibility gaps. The adaptation was deliberately limited to the FORECAST stage: REVIEW, SEARCH, and ACTION were not used for the scored static cards, and the same Brier/log extraction consumed only the returned yes probability.

I. Leakage and Scoring Notes

All reported Brier scores use $(p_{\text{yes}} - y)^2$, where $y = 1$ for a private yes resolution and $y = 0$ for a private no resolution. Log score is $-\log(\tilde{p}_{\text{yes}})$ for yes labels and $-\log(1 - \tilde{p}_{\text{yes}})$ for no labels, where $\tilde{p}_{\text{yes}}$ is clipped to $[0.01, 0.99]$. For WorldFork rows, unresolved mass can contribute neutrally to the extracted binary probability, but it remains an audit quantity and should be reported separately from the binary score. The scoring loop is:

```
for each resolved case:
  read the frozen public prediction
  read the private label y in {0, 1}
  extract p_yes for the method
  compute Brier = (p_yes - y)^2
  compute clipped binary log score
  for WorldFork:
    settle each path as yes, no, or unclear
    p_yes = yes_mass + 0.5 * unclear_mass
    report unresolved_mass separately
```

J. Per-card Predictions

Table 11 reports the per-card probabilities available from the direct baseline, DeepSeek structured baseline, ai-prophet, and WorldFork prediction artifacts. The WorldFork columns are assembled from available terminal path-mass exports and are used for the consistency checks in this draft. Values are rounded to four decimals; exact-looking entries such as 0.0000 and 1.0000 should be read together with the clipped log-score rule in Appendix D. A locked follow-up should export this table automatically before labels are opened. The blend column is the fixed equal blend of the GPT-5.4 structured direct probability and the WorldFork branch probability in this table.

Table 9. Available model and run identifiers. Rows without exact resolved snapshots should be treated as run labels, not immutable model definitions.

Row	Provider / route	Requested model	Resolved model string	Reproducibility note
Direct JSON GPT-5.4	OpenAI Codex	GPT-5.4	GPT-5.4	Snapshot date not available in the reported run label.
Direct JSON DeepSeek	OpenRouter	DeepSeek v4 Flash	deepseek-v4-flash-20260423	Resolved model string is available.
ai-prophet GPT-5.4	OpenAI	GPT-5.4	gpt-5.4-2026-03-05	Resolved model string is available.
WorldFork rollouts	OpenRouter + OpenAI Codex	DeepSeek v4 Flash cohorts; GPT-5.4-Codex non-cohort	WF mixed	Cohort agents used DeepSeek v4 Flash; initialization, global review, and other non-cohort roles used GPT-5.4-Codex/GPT-5.4.

Table 10. Leakage exposure summary for the reported rows. Masking reduces direct entity recall but cannot eliminate retrospective leakage.

Method	Full	IDs	Ret.	Local	Risk
Direct JSON GPT-5.4	yes	yes	no	no	med/high
Direct JSON DeepSeek	yes	yes	no	no	med/high
ai-prophet GPT-5.4	yes	yes	no	no	med/high
WF no-branch	decomp.	yes	no	yes	med
WF branching	decomp.	yes	no	yes	med

Table 11. Per-card probability table, cases 001–012. Direct is the structured GPT-5.4 direct row; DS is DeepSeek v4 Flash; Proph. is ai-prophet.

Case	y	Dir.	DS	Proph.	NoWF	BrWF	Blend
001	1	0.1035	0.1200	0.0800	0.2579	0.5591	0.3313
002	1	0.0709	0.2000	0.1200	0.2735	0.5579	0.3144
003	1	0.3854	0.3500	0.4200	0.2523	0.3630	0.3742
004	0	0.0151	0.3500	0.2800	0.7287	0.0001	0.0076
005	0	0.0133	0.3500	0.2400	0.1872	0.0007	0.0070
006	1	0.1616	0.1500	0.1600	0.9842	0.4872	0.3244
007	0	0.0659	0.4000	0.4400	0.1872	0.0000	0.0330
008	0	0.0857	0.5000	0.4900	0.3480	0.0002	0.0430
009	1	0.9776	0.5500	0.4300	0.2629	0.9989	0.9883
010	0	0.6473	0.6500	0.6100	0.7136	0.6407	0.6440
011	1	0.5177	0.6500	0.7200	0.2449	0.3103	0.4140
012	1	0.5717	0.3500	0.5800	0.9233	0.2622	0.4169

Table 12. Per-card probability table, cases 013–024. NoWF and BrWF are the no-branch and branching WorldFork probabilities.

Case	y	Dir.	DS	Proph.	NoWF	BrWF	Blend
013	0	0.0107	0.3500	0.4100	0.0475	0.0011	0.0059
014	0	0.8116	0.5500	0.6200	0.7262	0.5118	0.6617
015	1	0.4678	0.5500	0.5800	0.2530	0.1803	0.3240
016	1	0.8317	0.6500	0.8700	1.0000	0.0680	0.4499
017	1	0.9891	0.9500	0.9000	0.2622	0.9984	0.9938
018	1	0.0695	0.1500	0.2800	0.2652	0.5531	0.3113
019	1	0.5285	0.6500	0.7400	0.2522	0.3342	0.4314
020	0	0.0055	0.1500	0.1200	0.0972	0.4496	0.2275
021	0	0.0093	0.2500	0.3800	0.0345	0.0074	0.0083
022	0	0.0889	0.3500	0.3400	0.0079	0.0089	0.0489
023	0	0.0250	0.1500	0.1400	0.1872	0.0002	0.0126
024	0	0.0111	0.2000	0.2400	0.0000	0.0096	0.0103

K. Report and Audit Examples

These examples illustrate why audit metrics are reported separately from Brier. A forecast can be accurate but ungrounded, or inaccurate but diagnostically useful because the trace reveals which assumption failed.

Branching helps: resolved_001. The public card asks whether Candidate A wins a 2025 Nobel Prize. The true label is yes. In the available paired artifact, no-branch reports $p_{\text{yes}} = 0.2579$ while branching reports $p_{\text{yes}} = 0.5591$, reducing Brier by 0.3563. The useful branch behavior is not that the model knows the winner, but that alternate award-mechanism paths preserve more yes mass under the public source-packet evidence. A grounded report claim should cite the award endpoint, nomination/base-rate evidence, and the branch path that keeps the category-specific upset route alive; a claim that the result was already known before the deadline would be unsupported.

Branching hurts: resolved_016. The public card asks whether Phone A is available to customers by September 30, 2025. The true label is yes. The no-branch artifact reports $p_{\text{yes}} = 1.0000$, while the branching artifact reports $p_{\text{yes}} = 0.0680$. This is a useful audit failure: branching introduced or preserved too much contrary/uncertain mass despite a short-horizon company availability signal. Together with cases 006, 012, and 020, it illustrates the confident-correct dilution pattern in Appendix E. The trace should expose whether the error came from preorder-vs-availability confusion, over-weighted launch-risk branches, or missing authoritative source-packet settlement.

Blend helps: resolved_009. The public card asks whether Player C wins the 2026 Australian Open men's singles title. The true label is yes. The structured direct GPT-5.4 probability is 0.9776 and the WorldFork branch probability is 0.9989, yielding a fixed blend of 0.9883. This example shows complemen-

tarity: the scalar direct call is slightly more cautious, while branch mass gives more weight to a path in which the favorite/opponent-risk balance resolves positively. The report remains auditable only if the path summary and endpoint evidence state which tournament-path assumptions carried the additional yes mass.

Fixed-blend caveat. These examples are explanatory, not confirmatory. They were selected after inspecting the run artifacts, and the fixed blend was one of several explored comparisons. A locked follow-up should pre-register example-selection rules, primary metrics, and whether any blend is allowed before labels are opened.