World Drift or Question Drift? Question Identity as a Precondition for Continual World Model Evaluation

Shanglong Hu
The University of Sydney
Sydney, Australia
shhu7334@uni.sydney.edu.au

Abstract

Continual world models are commonly evaluated by tracking their predictions under fixed labels across model updates. We argue that such comparisons are meaningful only if the underlying question (what is predicted, under which conditions and within which horizon) stays the same. When this question changes while the label does not, we call it *question drift*. For example, a predicted collision risk can rise even as the environment becomes safer, simply because its deadline was extended. We show that a change in prediction splits exactly into world drift, question drift and a change in prediction error, and we prove that a log keeping only the current question definition may not even reveal whether the question changed. Question drift also distorts scoring: matched risk varies with event uncertainty even for a correct forecast, scoring revised predictions against the previous question's targets can overstate or understate error, and replacing historical questions hides forgetting. Controlled grid-world experiments with exact simulator probabilities illustrate each effect. We conclude with three recommendations for continual evaluation: log question definitions and evaluation protocols, score each prediction against its own target, and evaluate new checkpoints on fixed historical questions.

1 Introduction

Consider a robot whose world model predicts `collision_risk` at every planning step, and an evaluator who tracks this prediction across model updates (Figure 1). Between two evaluation rounds, the prediction rises from 0.45 to 0.68. The natural reading is that the environment became more dangerous or the model degraded. In fact, three things changed at once: the wall moved, the model was trained further, and the deadline behind the label was extended from 2 s to 4 s. The longer deadline accounts for most of the rise ($+0.28$, question drift) and the change in prediction error for $+0.04$; at the original deadline, the moved wall made the world safer (-0.09, world drift). The label stayed fixed while the question changed. A change in score speaks to learning, forgetting or environment change only if the question is held fixed.

The deadline is only one part of the question behind a prediction. A question also specifies which history to condition on, which actions or policy to apply, what target to predict, and how observations and actions are interpreted. Action-conditioned world models take history and actions as explicit inputs (Bar et al., 2025), and value functions also make goals (Schaul et al., 2015), policies, the predicted signal and its termination explicit (Sutton et al., 2011). When any of these components changes while the label stays the same, we speak of *question drift*.

Related work. Work on task specification separates the environment from the requested behavior (White, 2017). In continual reinforcement learning (Abel et al., 2023; Khetarpal et al., 2022),

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Workshop: Continual World Models.

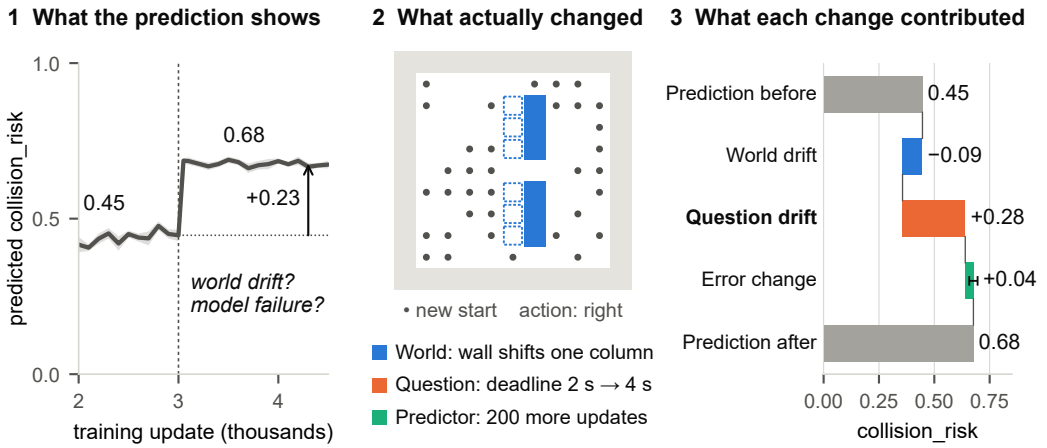


Figure 1: Question drift hidden in a continuing prediction. (1) The predicted `collision_risk` rises by 0.23 between two evaluation rounds. (2) Three changes occur together: the wall shifts one column, the deadline is extended from 2 s to 4 s under the same label, and the model receives 200 additional updates. (3) Exact simulator probabilities attribute the change; world and question terms average both attribution orders. Ten seeds, 35 new start cells (Appendix B); shading/error bar: ± 1 SE.

retention is measured by re-evaluating fixed historical tasks (Lopez-Paz and Ranzato, 2017). Dataset-shift and concept-drift methods detect changes in the joint distribution of inputs and targets (Gama et al., 2014; Moreno-Torres et al., 2012), and question drift can leave similar statistical signatures in an unchanged environment. Provenance-guided learning (Lamaakal, 2026) assumes that definition changes are recorded; we study questions that change silently under a persistent label.

We make three contributions. (i) We formalize question identity separately from the environment and the model, and prove that a log keeping only the current definition may hide a question change (Section 2). (ii) We decompose a change in prediction exactly into world drift, question drift and prediction-error change, and instantiate it with exact reference answers (Section 2). (iii) We identify three ways in which question drift biases evaluation and turn them into three recommendations (Sections 3–4).

2 Formalizing Question Drift

Setup. We treat a world model as a learned predictor: given past observations and planned actions or a policy, it outputs a distribution over future outcomes, or a statistic of that distribution, as in recurrent world models and Dreamer (Ha and Schmidhuber, 2018; Hafner et al., 2025) or Solaris (Savva et al., 2026). General value functions are a transparent special case (Schlegel et al., 2021). Let P denote the environment, q the question and θ the model parameters. The question fixes the components listed in Section 1, including the reported statistic. Two descriptions define the same question when these components are equivalent, as with deadlines of 2 s and 2,000 ms. Retraining or reimplementing the model leaves the question unchanged (Appendix A).

Evaluation protocol. We call the full specification behind a score its *evaluation protocol*: the questions with their semantics and weights, the environment, the evaluation distribution and the metric. Scores are directly comparable only within one protocol, so a revised question starts a new series of comparable scores.

Decomposing a change in prediction. Let r_t be the prediction at evaluation round t, made with parameters θ_t (here, the mean forecast over start cells), and let $V(P, q)$ be the exact answer over the same cells, so that the prediction error is $e_t = r_t - V(P_t, q_t)$. Adding and subtracting $V(P_1, q_0)$ gives the exact decomposition

$$r_1 - r_0 = \underbrace{V(P_1, q_0) - V(P_0, q_0)}_{\text{world drift}} + \underbrace{V(P_1, q_1) - V(P_1, q_0)}_{\text{question drift}} + \underbrace{e_1 - e_0}_{\text{error change}} . \tag{1}$$

Because each error is measured against its own environment and question, the error term can change even for a fixed model. Each term requires reference answers for the earlier question q_0 in both environments, or enough knowledge of q_0 to compute them. A log may not even reveal whether q changed: if only the current definition is kept, two deadline questions whose answers are zero on every logged history leave identical logs, even for an exact predictor (Proposition 1, Appendix A.2).

Instantiation. Figure 1 applies Eq. (1) to a stochastic grid world (Appendix B). The agent repeats one action from a start cell; each move succeeds with probability 0.8 and otherwise slips sideways, and the question asks how likely the agent is to hit a wall within the deadline. We train the model on sampled outcomes from 34 start cells, evaluate it on 35 new start cells and introduce all three changes by design; the simulator supplies exact reference answers. Over ten seeds, the prediction increases by 0.23 ± 0.02 (mean $\pm$ SE). The world and question terms are exact but depend on which change is attributed first, so we average both orders and obtain -0.09 and $+0.28$ (Table 1). The remaining $+0.04 \pm 0.02$ is the change in prediction error; Appendix B.3 splits it further and gives an alternative decomposition without exact answers.

3 Consequences for Evaluation

Matched risk depends on the question. The expected score of a forecast reflects its question as well as its accuracy. For a binary event B_q with probability $a_q(h) = \Pr(B_q = 1 \mid h, q)$ given history h and a fixed forecast $p_q(h) \in [0, 1]$, the expected Brier score (Brier, 1950) decomposes as (Gneiting and Raftery, 2007)

$$\mathbb{E}[(p_q(h) - B_q)^2 \mid h, q] = \underbrace{(p_q(h) - a_q(h))^2}_{\text{squared probability error}} + \underbrace{a_q(h)(1 - a_q(h))}_{\text{event uncertainty}}. \tag{2}$$

When a forecast is scored against the probability of its own question, its *matched target*, the first term is the excess risk and the sum is the matched risk. Since each question has its own event uncertainty, matched risk can change even when forecast error does not. In our grid world, with environment and parameters held fixed, matched risk rises and falls with event uncertainty as the deadline grows from 1 to 8 s, while excess risk falls from 0.040 to 0.003 (Appendix C.2). Even an exact predictor shows this pattern, with zero excess risk (Appendix A.4).

Mismatched targets overstate or understate error. Suppose a forecast for the revised question is scored against the previous question's target, which answers a different question. Figure 2 rescores the forecasts behind the later prediction in Figure 1 on the 35 new start cells, where they are imperfect. Each forecast p is compared with two exact targets in the same environment P_1: its matched target a_1 and the previous question's target a_0. Only the pairing differs. The squared probability error $(p - a_1)^2$ equals the excess Brier risk, whereas $(p - a_0)^2$ measures the distance to another question's answer. Their difference is $d^2 + 2ed$, where $d = a_1 - a_0$ is the gap between the targets and $e = p - a_1$ is the forecast error (Appendix A.3).

For the deadline revision from 2 s to 4 s, the error grows from 0.020 ± 0.002 under the matched target to 0.159 ± 0.008 under the previous question's target, and most of this increase is the target gap ($d^2 = 0.152$). The forecast error enters through $2ed$. It reverses the sign of the difference on 16% of per-cell scores, where the mismatched score understates the error. Other revisions show the same pattern, while both controls, which keep the question fixed, score the two pairings identically. Averaged over all 112 deadline pairs, the cross-term is only $+0.004$, yet 16% of per-cell scores are still understated. On training cells, where the model is nearly exact, the difference nearly equals d^2 (0.226 versus 0.227) and this fraction drops to 3% (Appendix C).

Replacing historical questions hides forgetting. Matching forecast and target to the current question avoids this bias but drops the original test from the evaluation. In a supporting experiment, we extend the deadline twice during training: at the end of each phase the loss on the current question is near zero, while the loss on the original question rises to 0.620. Fixed-question evaluation (Lopez-Paz and Ranzato, 2017) exposes this standard catastrophic forgetting, which current-question scoring hides (Appendix D).

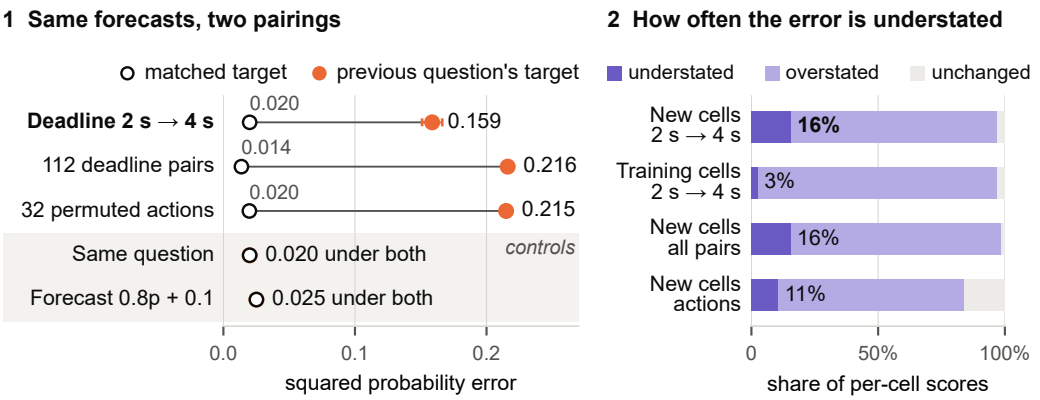


Figure 2: Target pairing changes the measured error of the same forecasts (later prediction of Figure 1; ten seeds; Appendix C). (1) Squared probability error against the matched (open) and the previous question's (filled) target; error bars: ± 1 SE. The 112 deadline pairs comprise 4 actions $\times$ 28 lengthened deadlines; action reinterpretations permute each action's meaning under its label; the two controls keep the question (original forecasts; forecasts set to $0.8p + 0.1$). (2) Fraction of per-cell scores (cells $\times$ seeds) that the previous question's target understates, overstates or leaves unchanged.

4 Recommendations for Continual Evaluation

R1: Log questions and evaluation protocols. Store the question, environment, evaluation distribution, metric and weights with every score, and flag each revision; undocumented protocols already hinder Atari comparisons (Machado et al., 2018). For binary events, also report excess risk.

R2: Score each forecast against its own target. Otherwise, the target gap counts as error.

R3: Evaluate new checkpoints on fixed historical questions. Report retention on fixed questions alongside current performance (Kearney et al., 2023; Liu et al., 2025; Lopez-Paz and Ranzato, 2017). Saved predictions and targets suffice to recompute an old score, even if the old question is unknown (Appendix A.2); testing a new checkpoint needs more (Appendix F):

Goal	Information needed
Recompute an old score	Saved predictions and targets, metric, weights
Rerun an old test	Historical inputs, target construction and scoring
Identify the full question	Definitions that distinguish candidate meanings

5 Limitations and Conclusion

Our study has three limitations. All evidence comes from grid worlds with exact simulator probabilities; elsewhere, Eq. (1) requires saved answers to the earlier question. Deciding whether two descriptions define the same question can be hard when sensors, action semantics or software change. Finally, we use one model family and two environments, and the studies in Appendices D and E evaluate on training states, so they show that each mechanism can occur, leaving its frequency open. Next steps are video world models and recovering historical questions across updates.

Labels can persist while questions change. Before reading a changed world-model score as learning, forgetting or world drift, evaluators should check that the same question was scored.

Acknowledgments and Disclosure of Funding

The author received no external funding for this work and declares no competing interests.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/9d8cf1247786d6dfeefeeb53b8b5f6d7-Abstract.html.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Bar_Navigation_World_Models_CVPR_2025_paper.html.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014. doi: 10.1145/2523813.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pages 2450–2462, 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Alex Kearney, Anna J. Koop, and Patrick M. Pilarski. What's a good prediction? Challenges in evaluating an agent's knowledge. *Adaptive Behavior*, 31(3):197–212, 2023. doi: 10.1177/10597123221095880.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022. doi: 10.1613/jair.1.13673.

Ismail Lamaakal. Provenance guided incremental learning under evolving concept definitions. *arXiv preprint arXiv:2608.23893*, 2026. URL https://arxiv.org/abs/2608.23893.

Zichen Liu, Guoji Fu, Chao Du, Wee Sun Lee, and Min Lin. Continual reinforcement learning by planning with online world models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 38397–38423, 2025. URL https://proceedings.mlr.press/v267/liu25p.html.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 6467–6476, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018. URL https://jair.org/index.php/jair/article/view/11182.

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012. doi: 10.1016/j.patcog.2011.06.019.

Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4): 595–600, 1973. doi: 10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2.

Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in Minecraft. *arXiv preprint arXiv:2602.22208*, 2026. URL https://arxiv.org/abs/2602.22208.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1312–1320, 2015. URL https://proceedings.mlr.press/v37/schaul15.html.

Matthew Schlegel, Andrew Jacobsen, Zaheer Abbas, Andrew Patterson, Adam White, and Martha White. General Value Function Networks. *Journal of Artificial Intelligence Research*, 70:497–543, 2021. doi: 10.1613/jair.1.12105.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, volume 28, pages 2503–2511, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html.

Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 761–768, 2011. URL https://sites.ualberta.ca/~pilarski/docs/papers/Sutton_2011_Horde_AAMAS.pdf.

Martha White. Unifying task specification in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3742–3750, 2017. URL https://proceedings.mlr.press/v70/white17a.html.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable memory for video world models. *arXiv preprint arXiv:2608.07408*, 2026. URL https://arxiv.org/abs/2608.07408.

A Questions, answers and the identification boundary

A.1 Questions, answers and implementations

Write $q = (\kappa, u, Y, \tau, \iota)$, where κ selects the history a forecast conditions on, u specifies actions or a policy, $Y = (Z, F)$ gives the target Z and the statistic F reported from its distribution, τ fixes temporal scope, and ι interprets tokens, sensors and actions. For a history-valued random variable H_t and a conditioning history $h \in \mathcal{H}$, the requested distribution and answer are

$$\mathfrak{L}(P, q; h) = \mathcal{L}_P^{u,\iota}(Z_q \mid \kappa(H_t) = \kappa(h)), \qquad \mathcal{A}(P, q; h) = F(\mathfrak{L}(P, q; h)). \tag{3}$$

The intervention u and interpretation ι determine how requested actions are applied. F may return a distribution, an expectation or an event probability; for an event probability we write $a_q(h) = \mathcal{A}(P, q; h)$, and for a fixed history distribution μ the answer in Eq. (1) is $V(P, q) = \mathbb{E}_{h\sim\mu}\, \mathcal{A}(P, q; h)$.

Two specifications denote the same question when their components are semantically equivalent, for example after a fixed unit conversion; syntactic equality is not required, and equal answers on a finite set of histories are not sufficient (Appendix A.2). An implementation update that preserves the specification retains the question, although learned predictions may change. So do memory updates (Wu et al., 2026) that leave the declared meaning fixed. Requested actions belong to q, and how actuators respond belongs to P (White, 2017). Executing eight seconds for a two-second question is an execution error rather than a new question; configuration and dependency debt make such errors common (Sculley et al., 2015). Establishing equivalence for physical grounding or general programs needs evidence beyond the record.

A general value function (Schlegel et al., 2021) is an instance: u supplies policy π, Z_q is the return defined by cumulant c and continuation γ, F takes expectation, and τ fixes when the return terminates.

A.2 Records that cannot reveal a question change

Two deadline questions can give the same answer at every logged history. If only the current definition survives, the record then cannot show which question was asked earlier. The limit is one of missing information, not of computation (the result is not about undecidability), and recovering the historical definition or its dependencies removes it.

Proposition 1 (Question identity is not identifiable from incomplete records)**.** *If the available record is consistent with both a fixed and a revised question, no decision rule using only that record can always determine whether the two predictions answer the same question. This can hold with a fixed environment and an exact predictor.*

Construction. The environment has two possible start histories. From h_{clear}, no collision occurs by step 8. From h_{obstacle}, collision occurs at step 4. Let q_2, q_8 request the probability of any collision by steps 2 and 8, respectively, with the other specification components equal. An exact predictor returns zero for both at h_{clear}, but returns zero and one, respectively, at h_{obstacle}. Thus the specifications are distinct even though their answers coincide at the logged history.

The retained record contains h_{clear}, two predictions equal to zero under the same label, the current q_8 definition, and the same exact-predictor version. It contains no historical question definition, semantic ID, defining hash, resolved dependency or external historical benchmark definition. In construction S (same), the old question was q_8; in construction R (revised), it was q_2. Every retained field is identical. The same-question claim is true in S and false in R. A deterministic rule must return the same decision on both records and so cannot always be correct. Nor can randomization guarantee correctness in both constructions. A rule may abstain, but abstaining does not settle whether the question changed. □

An executable script confirms that the serialized records are identical, that the same-question claim has opposite truth values, and that the questions differ at h_{obstacle}. Recovering the historical definition resolves the ambiguity.

Question identity and score recovery. Appending a saved zero collision target at h_{clear} to both records preserves their equality and the opposite truth values of the same-question claim. The saved prediction and target give Brier loss zero in both constructions. Thus historical identity can

remain ambiguous while a score on the logged evaluation set is computable. A historical interface that preserves the test inputs, conditioning history, target construction and scoring can also support evaluating later checkpoints on that set. Identifying the full question requires information that separates the candidate meanings. Recomputing a saved score uses the prediction–target pairing, scoring rule and aggregation weights; it need not recover the full definition.

A.3 Pairing effects for outcomes and for probabilities

Realized outcomes. For the same forecast p scored against two realized binary targets, define the pairing effect

$$\Delta\ell = (p - y_0)^2 - (p - y_1)^2 = d^2 + 2ed, \qquad d = y_1 - y_0, \quad e = p - y_1.$$

The first term measures target disagreement; the cross-term depends on the forecast error and its alignment with that disagreement. For exact deterministic forecasts $e = 0$, and the mean effect equals the changed-target fraction; for imperfect forecasts either sign is possible. For $p \in [0, 1]$ and binary targets, $-d^2 \le \Delta\ell \le d^2$, so the absolute mean effect is bounded by the target-disagreement fraction, and unchanged targets give zero effect for every forecast. With $(y_0, y_1) = (0, 1)$, forecasts $p = 0.1, 0.5, 1$ give effects $-0.80, 0, 1$, respectively. A negative effect means that the mismatched score is lower than the matched one.

Probability targets. With exact event probabilities a_0, a_1 in place of realized outcomes, the same identity holds for the squared probability error $(p - a)^2$. Against the matched probability a_1 it is the excess Brier risk of the forecast (Eq. (2)); against a_0 it only measures the distance to a target the forecast was not meant for and is not an excess risk. Expected Brier risk against realized outcomes adds the event uncertainty $a(1 - a)$ of the target used. Appendix C measures both signs on forecasts with genuine error.

A.4 Question-dependent risk under correct scoring

In Eq. (2), the squared probability error is excess Brier risk; the event-uncertainty term is the optimum risk (Gneiting and Raftery, 2007). Averaged over histories it equals the base-rate uncertainty $\bar{a}_q(1 - \bar{a}_q)$ of Murphy's partition (Murphy, 1973) minus $\mathrm{Var}_h\, a_q(h)$, with $\bar{a}_q = \mathbb{E}_h\, a_q(h)$. This term changes with the event even under correct scoring; it stays fixed only if the question, the world and the history distribution stay fixed.

An exact construction keeps one history, policy and trajectory distribution unchanged. The collision time is one second with probability 0.01, four seconds with probability 0.49, and infinity with probability 0.50. Collision-by-two and collision-by-eight therefore have probabilities 0.01 and 0.50. Exact predictors answer both questions at both checkpoints. Enumerating all three outcomes with rational arithmetic gives $R_2 = 0.0099$ and $R_8 = 0.25$, each equal to its optimum risk $R_q^* = a_q(h)(1 - a_q(h))$. A correctly matched score that follows the changed deadline increases by 0.2401 with zero change in excess risk. Two independent enumerations of the saved analysis check these values.

A.5 Attribution order and the error term

A change of the prediction from 0 to 1 can have three different causes: an exact predictor switches between two questions with outcomes 0 and 1; the world changes so that a fixed question's outcome goes from 0 to 1; or a fixed question with answer 0 receives an estimate of 1. If the observed history is the same and the requested interventions were not logged, the three cases leave identical records; controls or complete logs distinguish them.

Eq. (1) passes through the intermediate pair (P_1, q_0). Passing through (P_0, q_1) instead moves the interaction $V(P_1, q_1) - V(P_1, q_0) - V(P_0, q_1) + V(P_0, q_0)$ from the question term to the world term and leaves the error term unchanged; Figure 1 and Table 2 report the mean of the two orders. The identity is an accounting rule: each term needs a reference answer for the earlier question. The error term is not a pure parameter effect, because a fixed predictor's error can change under another world or question; the decomposition by checkpoint and question in Appendix B.3 separates the model update instead.

B Decomposition experiment

B.1 Environment, predictor and schedule

Environment. The grid is 11×11 with boundary walls. In world P_0 an internal wall occupies column 5, rows 2–8, with an opening at row 5; world P_1 moves the wall to column 6. An intended move succeeds with probability 0.8; otherwise the agent moves to one of the two perpendicular directions with probability 0.1 each. An attempted move into a wall is a collision; the agent stays in place and the episode continues. A question fixes a start cell, an action repeated at every step and a deadline $\tau \in \{1, \ldots, 8\}$ steps (one step stands for one second); its event is at least one collision by τ. A survival recursion over cells and remaining steps gives exact event probabilities.

Predictor. The 69 cells free in both worlds are split once into 34 training and 35 new start cells; the predictor never trains from a new start cell. It receives normalized coordinates, the action and the deadline, but not the world. Three 128-unit ReLU layers feed a sigmoid collision output trained by cross-entropy on sampled outcomes: each example simulates one trajectory from a training cell under one of the 32 questions (four actions, eight deadlines). Adam uses learning rate 0.002 and batch size 128.

Schedule. Each run trains 3,000 updates in P_0, then continues training in P_1 without being told that the world changed; predictions are read after $K \in \{0, 50, 200, 1000, 3000\}$ further updates, with $K = 200$ fixed in advance as the primary budget. A three-seed pilot without the world change confirmed trainability before the main runs (training-cell error 0.049–0.058, criterion < 0.10).

B.2 Decomposition and design history

The prediction r is the mean forecast over the new start cells: prediction 0 uses (P_0, q_0, θ_0) after 3,000 updates and prediction 1 uses (P_1, q_1, θ_K). With exact answers $V(P, q)$, the exact event probability averaged over the same cells, the change splits as in Eq. (1) with either the world or the question attributed first; the two orders differ by the interaction (Appendix A.5), and Figure 1 shows their mean. The prediction-error change $[r_1 - V(P_1, q_1)] - [r_0 - V(P_0, q_0)]$ does not depend on the order. World and question terms are exact functions of the designed changes; predictions and the error term are estimated.

The plan first fixed the deadline pair $2 \to 8$ steps (E1, seeds 300–309). At 8 steps a collision is nearly certain from most new start cells (exact probability 0.97–1.00 in both worlds), so moving the wall barely changes the answer and the world term nearly vanishes when the question is attributed first; this ceiling, not the predictor, limited E1. A second prospective run (E1b) therefore changed only the second deadline to 4 steps, chosen because its exact probabilities stay below 0.8 in both worlds, and used fresh seeds 310–319 because E1's error values were already known. E1b is the primary result; Table 1 reports both, and Table 2 gives the full deadline map declared before E1b ran.

B.3 Decomposition by checkpoint and question (reanalysis)

The world is not an input to the predictor, so a prediction depends only on the checkpoint and the question. Two single-change arms, specified in the redesign plan and run with E1b, complete the predictions of Table 3; this decomposition needs no exact answers and is a reanalysis of saved results. It answers a different question from Eq. (1): it attributes the change in prediction to changing the question and to further training, not to the world and the prediction error. Through the old checkpoint, changing the question moves the prediction by +0.315 and further training then by -0.084; through the old question, further training moves it by -0.073 and changing the question then by +0.303. The orders differ by an interaction of -0.011.

The same four predictions also decompose the prediction-error change of Eq. (1). Read before any update in P_1 ($K = 0$), it is +0.121 $\pm$ 0.021, an error carried over from P_0: the predictor still forecasts the old wall. The 200 updates remove 0.084 of it, which equals the further-training step at the new question, leaving +0.037. On training cells at $K = 200$ the prediction-error change is +0.005 $\pm$ 0.016, so the +0.037 on new start cells mainly reflects prediction 0's larger underestimate (0.446 against an exact 0.495).

Table 1: Decomposition on the new start cells after $K = 200$ updates; mean $\pm$ SE over ten seeds, exact answers in parentheses. World, question and interaction terms are exact.

Quantity	E1b: 2 $\to$ 4 steps	E1: 2 $\to$ 8 steps
Prediction 0 (exact answer)	0.446 $\pm$ 0.019 (0.495)	0.435 $\pm$ 0.013 (0.495)
Prediction 1 (exact answer)	0.677 $\pm$ 0.010 (0.688)	0.956 $\pm$ 0.004 (0.934)
Observed change	+0.230 $\pm$ 0.019	+0.521 $\pm$ 0.013
World drift, world / question first	$-$0.096 / $-$0.084	$-$0.096 / $-$0.004
Question drift, world / question first	+0.289 / +0.277	+0.535 / +0.443
Interaction	+0.012	+0.092
Prediction-error change	+0.037 $\pm$ 0.019	+0.082 $\pm$ 0.013

Table 2: E1b deadline map from 2 to τ_1 steps (new start cells, $K = 200$). World and question terms are the mean of the two orders.

τ_1	Exact $V(P_1, q_1)$	Change	World	Question	Error change
3	0.569	+0.094	$-$0.087	+0.162	+0.020 $\pm$ 0.018
4	0.688	+0.230	$-$0.090	+0.283	+0.037 $\pm$ 0.019
5	0.788	+0.348	$-$0.074	+0.368	+0.055 $\pm$ 0.017
6	0.851	+0.434	$-$0.063	+0.420	+0.077 $\pm$ 0.019
7	0.900	+0.470	$-$0.053	+0.458	+0.065 $\pm$ 0.019
8	0.934	+0.505	$-$0.050	+0.489	+0.065 $\pm$ 0.018

Table 3: Predictions by checkpoint and question (E1b, new start cells, $K = 200$); mean $\pm$ SE over ten seeds, exact answer in the checkpoint's training world in parentheses. Moving along a row changes the question; moving down a column adds 200 training updates in P_1.

Checkpoint	q_0: deadline 2 steps	q_1: deadline 4 steps
θ_0 (3,000 updates in P_0)	0.446 $\pm$ 0.019 (0.495)	0.761 $\pm$ 0.014 (0.772)
θ_K (+200 updates in P_1)	0.373 $\pm$ 0.017 (0.399)	0.677 $\pm$ 0.010 (0.688)

B.4 Verification

Exact probabilities were compared with 100,000-rollout Monte Carlo estimates for all 4,416 cell–question–world combinations and with training-label frequencies in 4,416 groups, with the criterion $|z| < 5$ for every comparison; none exceeds it, and the largest $|z|$ is 3.95. Every saved checkpoint reproduces its forecasts, and an independent script recomputes every term from the raw arrays. Code, forecasts and hashes are in the supplement and the weights in a separate archive.

C Target pairing and matched risk (reanalysis)

C.1 Design and results

This is a reanalysis of the saved E1b and E1 forecasts of Appendix B, planned before any calculation; no additional training or inference is performed. For a forecast p of question q_1, let a_1 be its exact event probability and a_0 that of the previous question q_0 in the same world. The squared probability error is $(p - a_1)^2$ against the matched target and $(p - a_0)^2$ against the previous question's target; the first is the forecast's excess Brier risk, the second only a distance to a target the forecast was not meant for. Their difference is $d^2 + 2ed$ with $d = a_1 - a_0$ and $e = p - a_1$ (Appendix A.3); a per-cell score is overstated if the difference is positive and understated if it is negative. Each comparison uses one checkpoint in one world, so only the pairing changes. The primary setting is the later prediction of Figure 1: E1b, world P_1, $K = 200$, new start cells. The 112 deadline pairs combine the four actions with the 28 pairs in which the deadline lengthens. Action reinterpretation keeps the action label of each of the 32 questions but cycles its meaning up $\to$ down $\to$ left $\to$ right $\to$ up; the predictor-change control keeps the question and raises the error by 0.005 under both pairings.

For the primary pair, expected Brier risk against realized outcomes, which adds event uncertainty, is 0.123 with the matched target and 0.256 with the previous question's target. The pattern is the

Table 4: Squared probability error of the same forecasts against the matched target and the previous question's target (primary setting; mean ± SE over seeds 310–319). Percentages are computed over all cells and seeds.

Case	Matched	Previous question	d^2	$2ed$	Over / understated
Deadline, right 2 → 4 steps	0.020 ± 0.002	0.159 ± 0.008	0.152	−0.013	81% / 16%
All 112 deadline pairs	0.014 ± 0.00038	0.216 ± 0.002	0.199	+0.004	83% / 16%
32 action reinterpretations	0.020 ± 0.00045	0.215 ± 0.003	0.223	−0.028	73% / 11%
Same question (control)	0.020 ± 0.002	0.020 ± 0.002	0.000	+0.000	0% / 0%
Predictor $0.8p + 0.1$ (control)	0.025 ± 0.002	0.025 ± 0.002	0.000	+0.000	0% / 0%

Table 5: Correctly matched risk by deadline (action right, primary setting; ten seeds). Brier risk is the sum of event uncertainty and excess risk.

Deadline (steps)	Brier risk	Event uncertainty	Excess risk
1	0.090	0.050	0.040 ± 0.007
2	0.130	0.097	0.033 ± 0.002
3	0.135	0.113	0.022 ± 0.002
4	0.123	0.103	0.020 ± 0.002
5	0.123	0.108	0.015 ± 0.001
6	0.106	0.095	0.011 ± 0.001
7	0.080	0.074	0.006 ± 0.001
8	0.056	0.054	0.003 ± 0.00036

same for other budgets K, for world P_0 before the change and for the E1 seeds: the previous-minus-matched difference is 0.126–0.171 and 12%–19% of per-cell scores are understated. On training cells, where forecasts are nearly exact, the difference is 0.226 against target disagreement 0.227, and 3% are understated. For action right, the difference grows with the deadline gap, from 0.007 (7 → 8 steps) to 0.695 (1 → 8 steps); the supplement lists every pair, action and setting.

C.2 Matched risk across deadlines

With world, checkpoint and new start cells fixed, only the deadline changes (Table 5). Matched risk rises and falls with event uncertainty; excess risk is smaller and falls as the deadline lengthens, because long-deadline events approach certainty. Over all 32 questions, excess risk lies between 0.003 and 0.044 and event uncertainty between 0.045 and 0.146.

C.3 Verification

Inputs are checked against the hashes recorded by the E1 analyses, the identity $(p-a_0)^2-(p-a_1)^2 = d^2 + 2ed$ holds on every cell, and an independent script recomputes all 1,182 reported entries from the raw arrays with explicit per-cell loops (largest difference below 10^{-15}).

D Supporting study: forgetting hidden by a changing scored question

This study does not show that question drift causes forgetting. It shows an evaluation failure: when the score follows the current question, deterioration on a historical question can stay invisible, and a fixed-question evaluation matrix reveals it by holding the target constant across checkpoints. Evaluation uses the states seen in training, so the study concerns this evaluation mechanism, not generalization to new states.

Following the fixed-task evaluation matrix of continual learning (Lopez-Paz and Ranzato, 2017), write the risk of checkpoint t on historical question q_i as

$$R_{t,i} = \mathbb{E}_{h\sim\mu_{\mathrm{ref}}}\mathbb{E}_{z\sim\mathfrak{L}(P_{\mathrm{ref}},q_i;h)}\ell\Big(\widehat{\mathcal{A}}_{\theta_t}(q_i;h), z\Big)\,, \tag{4}$$

with collision outcome $z \in \{0,1\}$, its Bernoulli distribution $\mathfrak{L}$, forecast probability $\widehat{\mathcal{A}}_{\theta_t}$ and $\ell(p,z) = (p-z)^2$. With q_i, P_{ref}, μ_{ref} and ℓ fixed, differences within a column reflect only the predictor.

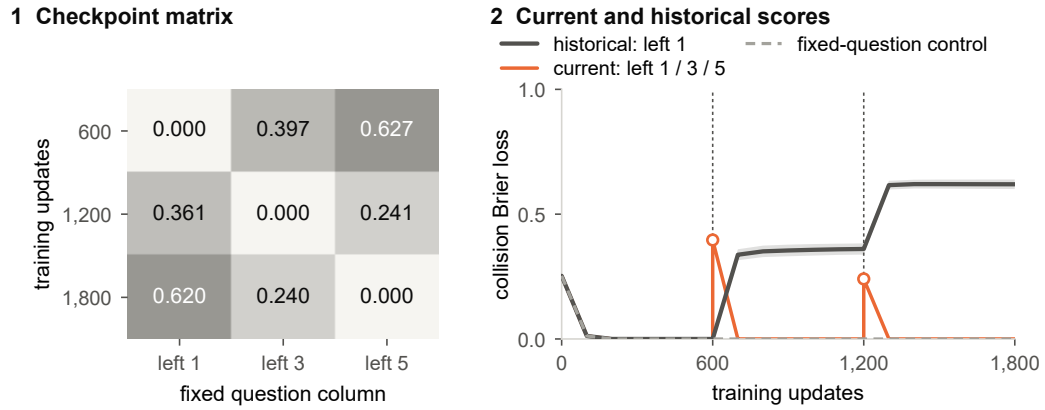


Figure 3: Forgetting hidden by a changing scored question (training states). (1) Each column fixes one left-action deadline; each row is a trained checkpoint. Entries average collision Brier loss over 75 states and ten seeds; the diagonal is the current-question score at the end of each phase. (2) Loss on the original one-step question rises while the current-question score at the end of each phase stays near zero. Open orange markers evaluate the next deadline at the preceding checkpoint, before any parameter update, so these jumps come from changing the question alone. Bands show ± 1 SE over seeds.

Design. The plan and primary contrasts preceded training. The network, loss and optimizer settings are those of Appendix E; the network receives the full action sequence and deadline, training uses no replay, and optimizer state carries across phases. The deterministic world, all 75 free states (used for training and evaluation), the left action, the any-collision target and the Brier score stay fixed; only the deadline changes. Ten seeds (200–209) each run a deadline stream (left 1, left 3 and left 5 steps for 600 updates each) and a fixed-question control (left 1 for 1,800 updates) from identical weights and sampling streams; the two arms coincide through the first phase. At update zero and every 100 updates the model answers all three questions on all 75 states. All 380 checkpoints, with weights, optimizer state, forecasts and targets, are archived; no seed, checkpoint or outcome is excluded.

Results. At updates 600 and 1,200 the next question is also evaluated with the pre-update weights: changing the question alone raises matched risk by 0.397 ± 0.001 and 0.241 ± 0.00043. Subsequent updates learn the new question, so end-of-phase scores omit these jumps, which Figure 3 retains. End-of-phase loss on the original left-1 question is 1.1×10^{-7}, 0.361 ± 0.023 and 0.620 ± 0.019 after 600, 1,200 and 1,800 updates, while end-of-phase current loss stays below 1.2×10^{-7}. Every seed's original-question loss rises between updates 600 and 1,800 (range 0.448–0.641), and the fixed-question control finishes at 1.1×10^{-9}, so the rise is measured on the same test rather than inferred from different questions. With the environment fixed, this is ordinary catastrophic forgetting. The fixed-question matrix reveals it; a score that switches to the new question does not. Every definition is retained, so the full matrix stays evaluable, unlike the record of Appendix A.2.

Verification. The archive also keeps previous-question scores of every current forecast and a frozen update-600 evaluator that separates the question change from later learning. An independent simulator reconstructs every target, and reloading all 380 weight files reproduces both prediction heads exactly.

E Questions executed through a controlled grid interface

Fitted models. The grid has boundary walls and an internal wall at column 5, rows 2–8, with an opening at row 5, leaving 75 free states. Twelve questions repeat up, down, left or right for 1, 3 or 5 steps. Inputs concatenate a one-hot state, five padded one-hot action slots and a one-hot horizon; two 128-unit ReLU layers feed a categorical terminal-state head and a binary any-collision head. Ten models (seeds 100–109) train with Adam (learning rate 0.002, batch size 96, 6,000 updates, terminal-state cross-entropy plus collision binary cross-entropy, single-threaded deterministic CPU) on uniform samples of the 900 state–question pairs and are evaluated on those same pairs. Mean training loss is 1.6×10^{-4} and collision Brier error 2.9×10^{-9}; terminal-state and collision classification error is

zero on these pairs. The archive keeps weights, predictions, targets, configurations and a replay audit against independently computed transition targets.

Execution protocol. Following a prospective plan, we load all ten archived models and submit five request batches per action over all 75 free states: the original one-step deadline, a three-step revision, the original expressed as 1,000 ms (one second per simulation step), a historical rerun after the revision, and an injected adapter fault that declares one step but executes three. The 200 batched calls produce 15,000 collision forecasts, 3,000 per case. The wrapper logs each call's declaration, resolved meaning, decoded action slots and horizon, and tensor and weight hashes.

Validation. A separate integer simulator constructs targets for the declared requests, and an independent checker rebuilds tensors and targets from the saved traces and recomputes every Brier score. Unit conversion and the historical rerun reproduce the original tensors and forecasts exactly; revision and fault calls return identical forecasts because they execute the same tensors under different declarations, and all 40 fault batches show the mismatch (Table 6).

Table 6: Executed-interface check on the fitted grid states. Mean collision Brier loss over 75 states, four actions and ten frozen models. The injected fault is scored against its declared one-step target.

Request	Declared/executed deadline	Declared-target loss	Original-target loss
Original	1 / 1	5.9×10^{-9}	5.9×10^{-9}
Deadline revision	3 / 3	1.3×10^{-9}	0.307
Unit equivalence	1 / 1	5.9×10^{-9}	5.9×10^{-9}
Historical rerun	1 / 1	5.9×10^{-9}	5.9×10^{-9}
Injected fault	1 / 3	0.307	0.307

Scope. The check covers a controlled wrapper on the fitted training states with one deliberately injected fault; it is not evidence about deployed systems or automatic detection. The supplement keeps the plan, all traces, forecasts and scores, the hashes, and a read-only replay of the 200 calls.

F Records for historical evaluation

Table 7 expands the three goals of Section 4 into the fields a record would keep; structured records, complete immutable logs or fixed external definitions can all hold them. A checksum shows that a file is unchanged, not what its values mean physically.

Table 7: Fields of a record for historical evaluation; each goal needs only some of them.

Field	Retained information
Semantic question	Immutable definition, lineage and the dependencies that fix its meaning; an identifier without recoverable meaning is not enough.
Implementation	Predictor and interface versions, dependency integrity, and any evidence that execution matches the definition.
Prediction context	Checkpoint, conditioning history or start state, predicted answer and environment reference where available.
Evaluation protocol	Intended goal; environment, evaluation distribution and weights, metric, and any substitution evidence.
Decision	Supported score, repeated-test result or identity assertion; otherwise the missing input and the conclusion it leaves open.

Example. A record for the later prediction in Figure 1 would contain the semantic question (version 2 of `collision_risk`: a collision within 4 s while the current action is repeated; it replaces version 1, whose 2 s definition is kept), the implementation (checkpoint θ_K and interface version), the prediction context (the 35 new start cells in world P_1), the evaluation protocol (equal cell weights and the Brier score) and the decision (the old score is recomputable, and version 1 can be rerun later).