

Fresh at Decision, Stale at Action: Routing New Information to LLM Decisions Already Made

Ayoub Merimi and Souhail Bakkali and Abdeljalil Sakat

Abstract

Freshness shapes an LLM decision at the moment it is made; the action it commits to is latent, carried out later, in a changing world. We therefore introduce a timestamp theory of decision validity and a decision ledger that realises it. Six timestamps define a decision; a freshness ceiling bounds what up-to-date evidence guarantees at action time, placing the remaining risk in information published afterwards. Accuracy then factorises into two capabilities: routing, which delivers that information to affected decisions before their revision deadlines, and temporal judgment, which reads it for the time of action. The ledger is persistent state holding each decision’s times, evidence and background knowledge, indexed by watch conditions written at decision time. Controlled decision streams with exact labels confirm four predictions across five LLMs, from 8B to frontier scale: freshness alone yields exactly the accuracy the ceiling predicts; models act on decisive news once it arrives, so accuracy follows routing; the ledger’s index sets what routing reaches, and watch conditions reach indirect dependencies at low exposure; and the remaining errors concern when a change takes effect. Freshness makes a decision right when it is made; routing and temporal judgment keep it right when it is carried out.

1 Introduction

At 13:05, a user asks an assistant for a walk-in dinner at 20:30. The assistant chooses Lovage Corner, the best-rated restaurant within walking distance, in the Bellhaven district. At 18:40, the grid operator announces emergency works at the Gorran substation from 19:50, leaving every area it supplies without power; the Gorran substation supplies the Bellhaven district (Figure 1). The choice was right at 13:05 and is wrong at 20:30. The information that settles it appeared after the decision and refers to the restaurant only through the substation

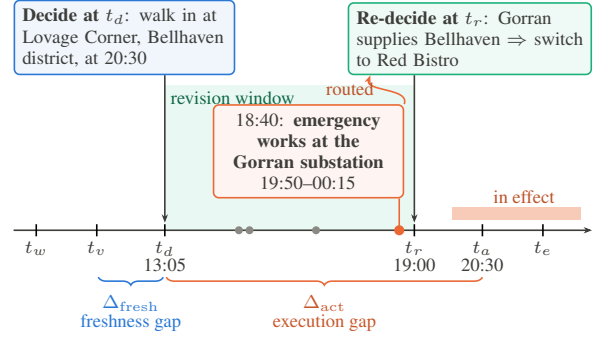


Figure 1: A decision from the INDIRECT stream on its six timestamps. Retrieval at decision time shrinks the freshness gap Δ_{fresh} . The notice that settles the decision is published inside the execution gap Δ_{act} , is in effect at action time, and names only the substation that supplies the restaurant’s district; background knowledge in the ledger routes it to the decision before the revision deadline. Grey dots are other notices in the window.

behind its district. This decision comes from our test streams (Appendix B.1).

The example separates two properties of a decision. *Evidence freshness* describes what was known when the choice was made. *Action validity* describes whether the choice is right when it is carried out. Research on time in language models secures the first: benchmarks track how models age (Lazaridou et al., 2021; Dhingra et al., 2022; Kim et al., 2024), test answers against evolving news (Liška et al., 2022; Kasai et al., 2023), and refresh models with search (Vu et al., 2024). This framing fits question answering, where an answer is used the moment it is produced. LLM agents produce decisions that are carried out later, such as itineraries (Xie et al., 2024b) and service actions (Yao et al., 2025), and these are judged by the second.

This paper develops a theory of action validity and answers the title question: freshness is necessary and insufficient. Six timestamps define a decision, from when its evidence is written to when its outcome is judged. A *freshness ceiling* shows

that, under a constant rate of change, the probability that evidence still holds when the user acts is bounded by the execution gap between deciding and acting, whatever the freshness at decision time. The remaining risk lies in information published inside that gap. Decision accuracy then factorises into *routing*, whether that information reaches the decision before its revision deadline, and *judgment*, whether the model reads it correctly for the time of action (Section 3).

The theory comes with a model of the system that realises it. A *decision ledger* is persistent state (Bakkali and Merimi, 2026) that stores each decision with its times, its evidence and its background knowledge, and a router delivers each new item to the open decisions it matches (Section 4). A query retrieves what the user asks about today; a ledger retrieves what yesterday’s decisions still need. The ledger also makes the theory testable: what a router can reach follows from what the ledger indexes, and what the model does with a delivered item follows from its judgment.

We derive four predictions and test them in controlled decision streams, in which every label follows from timestamps and stated facts, under three conditions that vary how news refers to a decision and how hard its negatives are: decisive news that names the option or its stated dependency (DIRECT), the same with hard negatives (STRESS), and news that reaches the option through other names and upstream entities (INDIRECT). The tests confirm all four predictions across five LLMs, from Ministral 8B to GPT-6.1 Sol.

- **P1, the freshness ceiling holds.** With perfect evidence at decision time and nothing afterwards, accuracy equals the share of decisions that should stand.
- **P2, delivery is sufficient for change.** Once decisive news arrives, models switch in 91–100% of cases under every router, so the accuracy of changing decisions tracks routing recall, and recall follows the ledger’s index: the decision text reaches 37.5% of news that uses another name for a dependency, and background knowledge in the ledger reaches all of it.
- **P3, misleading deliveries cost correct decisions.** Every misleading item delivered is a chance to abandon a decision that should stand. Watch conditions written at decision

time reach 92.2% of decisive news with 20 misleading deliveries against 89 for the whole background, and broadcasting helps only models that judge well.

- **P4, the residual error is temporal judgment.** Given exactly the related news, Ministral 8B and Qwen3-235B abandon every decision whose related change ended before the activity, Qwen3-8B abandons 94% of those whose change starts after it, and Claude Sonnet 4.6 keeps all of them.

This is a theory paper with controlled empirical tests. It contributes a temporal theory of LLM decisions with a freshness ceiling and a routing–judgment factorisation (Section 3), the decision ledger as a model of the system that realises it, including watch conditions (Section 4), and controlled tests that confirm the theory’s predictions across five models (Sections 5–7).

2 Related Work

Freshness and temporal knowledge. Language models degrade on text written after their training data (Lazaridou et al., 2021) and can be probed and refreshed for time-dependent facts (Dhingra et al., 2022; Kim et al., 2024). Benchmarks over evolving news and time-sensitive questions test whether answers are current (Chen et al., 2021; Liška et al., 2022; Kasai et al., 2023), search augmentation keeps them current (Lewis et al., 2020; Vu et al., 2024), and evaluation sets themselves go stale as facts change (Jiang et al., 2026). Temporal reasoning benchmarks test ordering and arithmetic over time (Fatemi et al., 2025). These works judge the model when it answers; we judge a decision when it is carried out.

Updates, conflicts and revision. Knowledge editing changes stored facts (Meng et al., 2022), and multi-hop and ripple-effect evaluations require an update to reach every fact that depends on it (Zhong et al., 2023; Cohen et al., 2024). Models resolve conflicts between parametric and contextual knowledge unevenly (Xie et al., 2024a; Xu et al., 2024), including conflicts that arise while agents act (Lyu et al., 2026), and revise conclusions under new premises with mixed success (Kassner et al., 2021; Wilie et al., 2024). Post-decision routing carries the ripple-effect requirement to decisions: a new fact must reach every open decision that

depends on it, and the model must then revise it correctly.

Agents that act over time. Agent benchmarks evaluate plans and actions executed after reasoning (Yao et al., 2023; Xie et al., 2024b; Yao et al., 2025). Time-aware simulations score how agents schedule work (Zhang et al., 2024), and agents misjudge when elapsed time calls for fresh tool calls (Cheng et al., 2026). Proactive agents decide when to intervene (Lu et al., 2025; Tang et al., 2026; Li et al., 2026; Xiaohongshu Dots Studio and Evolvent AI, 2026), maintain long-term intents through trigger conditions (Shi et al., 2026), and accept user corrections during execution (Zhai et al., 2026). We study many concurrent decisions with revision deadlines, a shared stream of world news, and the cost of deciding which news reaches which decision.

Memory and retrieval. Agent memories store and retrieve past interactions (Park et al., 2023; Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025), invalidate outdated facts in temporal graphs (Rasmussen et al., 2025), index memories by time and topic (Tian et al., 2026), and are evaluated on long-term recall (Wu et al., 2025; Du, 2026). Persistent, evidence-grounded state with temporal validity is a design requirement for systems that work across sessions (Bakkali and Merimi, 2026); the decision ledger applies it to commitments. Watch conditions are document expansion (Nogueira et al., 2019) and generated queries (Gao et al., 2023) written once, at decision time. BM25 remains a strong zero-shot retriever next to dense retrieval (Karpukhin et al., 2020; Thakur et al., 2021), and long contexts degrade reasoning when relevant content is buried or lacks literal overlap (Shi et al., 2023; Liu et al., 2024; Modarressi et al., 2025).

3 A Theory of Decisions Over Time

Six timestamps. Evidence is written at t_w and last verified at t_v . The model decides at t_d with evidence verified at $t_v \leq t_d$. The decision can be revised until the revision deadline t_r , is carried out at t_a , and its outcome is observed at t_e , with $t_d < t_r \leq t_a \leq t_e$. The revision deadline captures commitment: an order ships, a table is taken, a train departs.

The freshness ceiling. Let $s(t)$ be the state of the world and $U(a, s)$ the utility of action a in state s . The model chooses $\hat{a}$ at t_d , and the outcome is

$U(\hat{a}, s(t_a))$, so the target is the best action at t_a . The age of the evidence at action time decomposes as

$$t_a - t_v = \underbrace{(t_d - t_v)}_{\Delta_{\text{fresh}}} + \underbrace{(t_a - t_d)}_{\Delta_{\text{act}}}. \quad (1)$$

Retrieval and knowledge updates shrink the freshness gap Δ_{fresh} ; the execution gap Δ_{act} is set by the task. If a fact f changes at a constant hazard rate λ_f , it still holds τ time units after verification with probability $e^{-\lambda_f \tau}$, so

$$\Pr[f \text{ holds at } t_a] = e^{-\lambda_f \Delta_{\text{fresh}}} e^{-\lambda_f \Delta_{\text{act}}}. \quad (2)$$

Proposition 1 (Freshness ceiling). *For any fact f and execution gap Δ_{act} , the probability that f holds at action time is at most $e^{-\lambda_f \Delta_{\text{act}}}$, for every degree of freshness.*

Appendix A proves the proposition and extends it to decisions with several dependencies. Freshness is a property of the evidence; validity at action time is a property of the whole pipeline. The ceiling locates the residual risk in the execution gap and in the volatility of the evidence, and only information that arrives after t_d can reduce it: a notice published inside the execution gap that reaches the decision before t_r .

Post-decision routing. A ledger holds open decisions. Decision i has a chosen option a_i , alternatives, evidence E_i , times $t_d^i < t_r^i \leq t_a^i$ and an activity $[t_a^i, t_a^i + h_i]$. A stream delivers items x_j , each written and published at t_w^j . Item j is *decisive* for decision i if it is published in the revision window, $t_d^i < t_w^j < t_r^i$, concerns a_i or an entity a_i depends on, and blocks a_i over an interval I_j that overlaps the activity, so that a_i stops being the best option. An item about a_i or its dependencies that is not decisive is *related*: a change outside the activity, or news that leaves a_i usable. A router assigns each item to a subset of the decisions open at t_w^j , and at t_r^i the model reads E_i with the items routed to i and keeps or replaces a_i .

Routing and judgment. Let π be the share of decisions that should change. For those, let ρ be the routing recall, the probability that a decisive item reaches the decision; S the probability that the model switches when it does; and L the probability that it switches when it does not. For decisions that should stand, let K be the probability that the model keeps them. Final accuracy is

$$A = (1 - \pi) K + \pi [\rho S + (1 - \rho) L]. \quad (3)$$

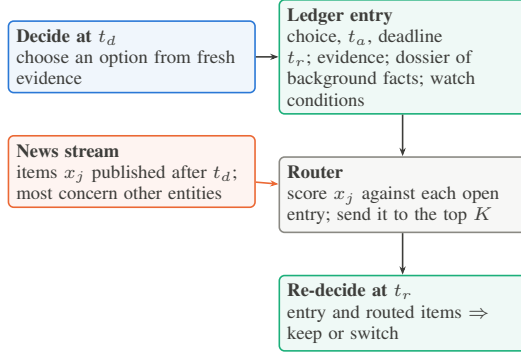


Figure 2: The decision ledger. Routing uses no model calls; the model is called again only for decisions that receive news. What the entry indexes sets what the router can reach.

Routing sets ρ and the items each decision sees; judgment sets S , L and K (Appendix A derives Eq. 3 and the trade-off between routers it implies). A routing failure leaves a decision stale; a judgment failure ignores decisive news or abandons a decision that should stand. A decisive item reaches the chosen option in one of four ways: it names the option (Lovage Corner), the dependency stated in the evidence (the Bellhaven district), another name for that dependency (Chime Green, as the district is known locally), or an entity upstream of it (the Gorran substation, which supplies the district). Each step down this list moves the link from the decision’s text into background knowledge.

Predictions. The theory makes four predictions. **P1:** without post-decision routing ($\rho = 0$, $L = 0$, $K = 1$), accuracy equals $1 - \pi$, the share of decisions that should stand, for every model. **P2:** delivery is sufficient for change ($S \approx 1$), so the accuracy of changing decisions tracks ρ , and ρ is set by whether the router’s index contains the entity a decisive item names. **P3:** each misleading item delivered lowers K at a rate set by the model’s judgment, so routers trade recall against exposure, and broadcasting, which maximises both, helps only models that judge well. **P4:** with oracle routing ($\rho = 1$, every related item and nothing else), the residual error concentrates on temporal judgment: deciding whether I_j overlaps the activity.

4 The Decision Ledger

Each ledger entry is persistent state that outlives the session in which the decision was made (Bakkali and Merimi, 2026). When the model decides at t_d , it writes the entry: the chosen option and its alter-

natives, the action time t_a , the revision deadline t_r , the evidence about the chosen option, and a *dossier* of background facts, such as other names of the entities the option depends on and what those entities rely on in turn. In deployment, the dossier holds what the assistant looked up while deciding, including much that is irrelevant to the user’s activity. When an item x_j is published, a router scores the open entries and sends the item to the K highest-scoring ones. At the revision deadline, the model reads the entry and the routed items in order of publication and keeps or switches. Decisions that receive no items keep their choice without a model call. Routers differ in what they index, which by P2 sets what they can reach.

Decision text. BM25 or dense embeddings compare the item with the decision’s request and its evidence about the chosen option, reaching news that names the option or its stated dependency.

Whole dossier. The decision text is expanded with every dossier fact, reaching news about other names and upstream entities together with news about everything else the dossier mentions.

Watch conditions. At t_d , the model reads the entry and writes three to eight conditions $w_{i1}, \dots, w_{ik}$ describing events that would stop the user’s activity. The conditions name every entity the activity depends on, including other names and upstream entities, and leave out what the user does not need (Appendix C). The router indexes the decision text with the conditions (BM25) or scores each decision by its best condition, $\max_k \cos(e(x_j), e(w_{ik}))$, with a sentence encoder e . Watch conditions are document expansion (Nogueira et al., 2019) performed once, at write time, by the component that knows why the option was chosen; by P3 they should match the dossier’s reach with less exposure.

Time filter. A rule drops an item for a decision when the item states a start time after the decision’s activity ends, performing by rule the temporal check of P4.

5 Controlled Decision Streams

Testing the predictions requires exact control over which items are decisive, how they refer to the option, and when changes take effect. Real logs offer none of this, and real entities let models answer from memory. We therefore generate fictional

decisions and news streams in which every label follows from timestamps and stated facts; the generator and data are released with the paper.

Decisions and labels. Each decision has two options. Option 1 is better at t_d (higher rating, lower price, faster journey, better views) and is the model’s initial choice; option 2 is acceptable but worse. Option 1 depends on an entity named in its evidence: a district, a rail operator, a sales or permit platform, a mountain range or a shopping centre. Lags between deciding and acting range from three hours to three days, and revision deadlines precede the action by 0.5 to 2 hours. A decision’s gold answer is option 2 if some item published in its revision window targets option 1, its dependency or an entity upstream of it and blocks option 1 during the activity, and option 1 otherwise. The generator excludes decisions invalidated before they were made and asserts every label (Appendix B).

Three conditions. Each condition stresses a different part of the theory. **DIRECT** measures every term of Eq. 3 with an open model; **STRESS** adds hard negatives that name the option, testing exposure (P3) and temporal judgment (P4); **INDIRECT** moves the reference from the decision text into background knowledge, testing whether the index sets recall (P2). **DIRECT** spans five domains with independent random news per decision: a decisive item ($p=0.5$) that names the option or its stated dependency, a late item ($p=0.3$) that blocks the option only after the activity, a harmless item ($p=0.5$), and two unrelated items. **STRESS** adds, for every decision, a *same-name decoy*, a blocking notice about a different entity whose name extends the option’s (“Clove Fork Annex”), and, each with probability 0.85, a late and a harmless notice naming the option itself; eleven rules paraphrase event wording. **INDIRECT** spans four domains and fixes every proportion. Half of the decisions should switch, and their decisive notice uses each of the four ways of reference for 16 decisions; only the first two match the decision text. The other half should stand and receive one of four hard negatives, each for 16 decisions: a change that starts after the activity, a change that is over before it starts, harmless news, and a disruption to a facility the user does not need (a tram line for a user who walks). Every decision receives exactly two related notices and one decoy, so neither the amount nor the wording of the news reveals the label. Each **INDIRECT** dossier holds 12 facts: the other name, the upstream entity,

	DIRECT	STRESS	INDIRECT
Decisions	110	75	128
should switch	60	48	64
via name or stated dependency	60	48	32
via other name or upstream	0	0	32
should stand	50	27	64
News items	396	458	512
decisive	56	48	64
related, not decisive	100	185	192
decoys and unrelated	240	225	256
Dossier facts per decision	—	—	12
Items in a decision’s window	115.7	128.2	78.8

Table 1: The three conditions. A decision’s revision window holds dozens to hundreds of items, of which a handful concern it.

four unneeded facilities, and six *traps* that mention entities belonging to other decisions in irrelevant roles (“Tram line T12 draws its power from the Hensall traction substation”). Table 1 summarises the conditions.

6 Experimental Setup

Routers. All routers of a condition share one re-decision prompt and differ in which items reach each decision. *None* routes nothing and realises perfect freshness at t_d . *All items* broadcasts everything published while the decision is open. *Name match* routes items that contain the option’s name. The routers of Section 4 send each item to its $K=3$ best-scoring open decisions; dense routing uses BGE-base (Xiao et al., 2024). *Oracle: related* delivers exactly the items about the option, its dependencies and its facilities, including every hard negative, and *Oracle: decisive* delivers only decisive items; both use gold annotations and isolate judgment, and *Oracle: decisive* also signals that a change is due, so it serves as a ceiling. **STRESS** also tests the time filter and *dependency queries*, three fixed queries naming the option and its stated dependency.

Models. On **DIRECT**, Qwen3-8B with thinking disabled (Yang et al., 2025) runs locally with greedy decoding (Wolf et al., 2020) and writes its own watch conditions. On **STRESS**, we evaluate Claude Sonnet 4.6, GPT-6.1 Sol and Qwen3-235B-A22B; on **INDIRECT**, Claude Sonnet 4.6, Qwen3-235B-A22B and Ministral 8B, with Mistral 7B (Jiang et al., 2023) reported in Appendix E. Hosted models run at temperature 0 with structured output, Claude writes the watch conditions for every model of a condition so that routing is identical across models, and every reported cell is complete

Router	Routing			Accuracy (%)		
	Items	Rec.	Tok.	Claude	GPT	Qwen
None	0.0	–	0.0	36.0	36.0	36.0
Name match	3.5	56	1.2	73.3	72.0	73.3
BM25: decision	18.0	100	2.0	97.3	98.7	85.3
BM25: + time filter	14.3	100	1.8	97.3	98.7	92.0
Dense: decision	18.0	96	2.0	90.7	96.0	85.3
Dense: watch cond.	18.0	96	2.0	93.3	97.3	86.7
Dependency queries	18.0	100	2.0	93.3	97.3	86.7
Oracle: related	3.3	100	1.2	96.0	100.0	90.7
Oracle: decisive	0.6	100	0.7	100.0	100.0	100.0
All items	128.2	100	8.1	96.0	100.0	78.7

Table 2: STRESS: final-decision accuracy (%) of Claude Sonnet 4.6, GPT-6.1 Sol and Qwen3-235B on the same 75 decisions. Items: new items delivered per decision; Rec.: share of decisive item–decision pairs delivered; Tok.: Claude Sonnet 4.6 input tokens per decision for the re-decision calls (thousands). Bold: best router without gold labels.

Router	Routing		Accuracy (%)		
	Recall	Traps	Claude	Qwen-235B	Ministral 8B
None	–	0	50.0	50.0	50.0
Name match	25	0	62.5	57.0	52.3
BM25: decision	62	17	78.9	60.2	55.5
BM25: + dossier	100	89	93.0	61.7	54.7
Dense: + dossier	86	59	87.5	60.9	57.8
BM25: + watch cond.	92	20	91.4	64.1	61.7
Dense: watch cond.	84	27	85.9	57.8	58.6
Oracle: related	100	0	99.2	70.3	59.4
Oracle: decisive	100	0	100.0	100.0	100.0
All items	100	96	91.4	56.2	56.2

Table 3: INDIRECT: final-decision accuracy (%) on 128 balanced decisions. Recall: decisive notices delivered to their decision (%); Traps: notices delivered to a decision whose dossier mentions their entity in an irrelevant role (of 96). Claude writes the watch conditions for every model. Bold: best router without gold labels.

with no imputed answers. Prompts and decoding settings are in Appendices C and D.

Measures. We measure accuracy A and the terms of Eq. 3: recall ρ , the switch rates S and L , and the keep rate K . *Traps* count items delivered to a decision whose dossier mentions their entity in an irrelevant role. Cost is input tokens, measured with Claude because prompts are identical across models. Exact McNemar tests compare routers on per-decision correctness, per model and pooled over the models of a condition; W–L counts decisions where the first router is right and the second wrong, and the reverse.

7 Testing the Predictions

Table 2 reports STRESS, Table 3 reports INDIRECT, and Appendix E reports DIRECT.

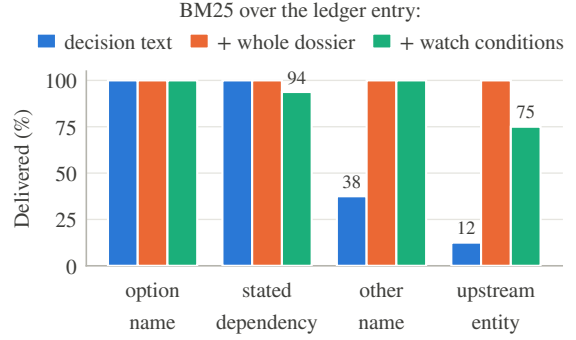


Figure 3: Decisive notices delivered on INDIRECT, by how the notice refers to the chosen option (16 decisions each; unlabelled bars reach 100%). The decision text reaches the option’s name and its stated dependency; the dossier and watch conditions extend the reach to other names and upstream entities.

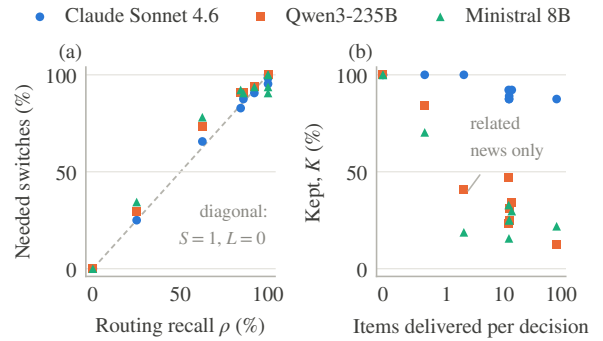


Figure 4: The two factors of Eq. 3 on INDIRECT, one point per router. (a) Needed switches against routing recall: on the diagonal, the model switches exactly when decisive news arrives. (b) Decisions kept among those that should stand, against the items delivered to them; the points at two items are the related-news oracle.

P1: the freshness ceiling holds. *None* has perfect evidence at decision time and nothing afterwards. Its accuracy equals the share of decisions that should stand for every model: 45.5% on DIRECT, 36.0% on STRESS and 50.0% on INDIRECT. Every decision that should change stays wrong, because the information that changes it was published after the decision. Routing any new information helps: name matching alone lifts Qwen3-8B on DIRECT to 70.0% ($p < 0.001$).

P2: delivery is sufficient for change, and the index sets recall. On INDIRECT, once a decisive notice reaches a decision, models switch in 90.6–100% of cases, for every router and every model. Needed switches therefore follow routing recall (Figure 4a): Claude’s points lie on the diagonal, within 4.7 points of recall under every router, and

wherever recall is below 100% the weaker models lie above it, because they also switch on decisions whose decisive notice never arrived. Accuracy on changing decisions is a routing quantity, and recall follows the index (Figure 3). Name matching reaches only news that names the option. The decision text reaches every notice that names the option or its stated dependency, 37.5% of those that use another name, and 12.5% of those that name an upstream entity, so Claude reaches 78.9%. Adding the dossier to the index reaches every decisive notice and lifts Claude to 93.0% (22–4, $p < 0.001$), and watch conditions lift it to 91.4% (21–5, $p = 0.002$). When news names the stated dependency, as on STRESS, the decision text is a sufficient index: BM25 delivers 100.0% of decisive items and lifts the three models from 72.0–73.3% with name matching to 85.3–98.7% (53–6 pooled, $p < 0.001$), as accurate as oracle routing of all related news for Claude (97.3% and 96.0%) and GPT (98.7% and 100.0%). Watch conditions, dense similarity and idealised dependency queries route as many items there, and pooled over models BM25 wins more discordant decisions against each (8–5, 11–4 and 6–3).

P3: misleading deliveries cost correct decisions.

Every item delivered to a decision that should stand is misleading, and Figure 4b shows its cost: Claude keeps at least 87.5% of these decisions under every router, while Qwen3-235B and Ministral 8B keep fewer than half under every BM25 and dense router. Indexing the whole dossier delivers 89 of the 96 trap notices, because every fact it holds becomes a match. Watch conditions deliver 92.2% of the decisive notices and 20 traps, close to the 17 of the decision text alone, because the model writes conditions only about what the activity needs. Under the retrieval and broadcast routers, a delivered trap raises the rate of needless switches for every model: from 9.4% to 14.0% for Claude, from 69.7% to 74.8% for Qwen3-235B and from 71.5% to 84.1% for Ministral 8B. Under retrieval routing, the weaker models also switch on 20–67% of the changing decisions whose decisive notice never arrived, prompted by other news, so their gains from recall come with losses on K . Lower exposure therefore matters most where judgment is weakest: watch conditions give Qwen3-235B and Ministral 8B their best accuracy and beat the whole dossier for Ministral 8B (12–3, $p = 0.035$), while for Claude the two indexes are equal (5–7,

Related news	DIRECT	INDIRECT		
	Qwen3-8B	Ministral	Qwen-235B	Claude
starts after the activity	6	0.0	31.2	100.0
over before it starts	–	0.0	0.0	100.0
harmless	89	50.0	68.8	100.0
facility not needed	–	25.0	62.5	100.0
all that should stand	62.0	18.8	40.6	100.0

Table 4: Judgment in isolation (P4): decisions kept (%) when the oracle delivers exactly the related news and none of it changes the best option. On INDIRECT, each row except the last covers 16 decisions per model.

$p = 0.77$). Broadcasting maximises both recall and exposure, at 4.0 times the tokens of BM25 routing on STRESS and 3.1 times on INDIRECT. Models that judge well absorb it (GPT 100.0% and Claude 96.0% on STRESS, Claude 91.4% on INDIRECT); the others lose: Qwen3-8B falls from 71.8% to 55.5% on DIRECT ($p = 0.003$), Qwen3-235B from 85.3% to 78.7% on STRESS, and on INDIRECT broadcasting loses to watch conditions (20–37 pooled, $p = 0.033$).

P4: the residual error is temporal judgment.

The oracles remove routing error (Table 4). Given only decisive items, every model on STRESS and INDIRECT is correct on every decision. Adding the related news that leaves the decision unchanged costs 91 decisions across the three INDIRECT models and gains none ($p < 0.001$), and 10 decisions to none on STRESS ($p = 0.002$). The errors concentrate on time. Ministral 8B and Qwen3-235B abandon every decision whose related change ended before the activity starts and keep 0.0% and 31.2% of those whose change starts after it, more often than they abandon decisions for harmless news or an unneeded facility; Qwen3-8B abandons 94% of decisions with a late change on DIRECT, although the prompt asks it to ignore such changes; Claude keeps all of them. Models react to the announcement of a change to their option and skip the check of when it is in effect, and stronger models skip it less often. Performing the check by rule confirms the diagnosis: on STRESS, the time filter raises Qwen3-235B from 85.3% to 92.0% (6–1, $p = 0.12$), while Claude and GPT stay at 97.3% and 98.7%.

8 Conclusion

Freshness is necessary for LLM decision-making and insufficient on its own. A decision is judged when it is carried out, the freshness ceiling bounds

what evidence at decision time can guarantee, and the accuracy that remains is set by routing and judgment (Eq. 3). The controlled tests confirm each prediction: freshness alone leaves changing decisions stale, delivery is sufficient for change and recall follows what the ledger indexes, misleading deliveries cost correct decisions in proportion to weak judgment, and the residual error is mostly temporal. The theory yields three design principles for LLM agents: *record* each decision with its times and what it depends on; *route* news to it until its revision deadline, through an index written when the decision is made; and *revalidate* it by checking when each change is in effect.

Limitations

Controlled streams. The decision streams are synthetic by design: exact control over timing, reference and labels is what lets each prediction be tested in isolation. Real news is longer and noisier, dependencies can be implicit, and relevance can be uncertain, so the magnitudes we report belong to these streams while the predictions concern mechanisms. Testing the theory on logged agent decisions, and in languages beyond English, is the next step.

Scope of the theory. The theory models a decision as a choice among options with one revision deadline and a binary outcome, and the freshness ceiling assumes a constant hazard rate. Real commitments are staged, partly reversible and graded in utility, and facts change in bursts. The factorisation of Eq. 3 carries over, because information must reach a decision before it can change it, but its terms then depend on time and on the cost of revising.

Reach of the ledger. Routing recall is bounded by the dependencies the ledger records at decision time, and the ledger treats routed news as trustworthy. Discovering dependencies after the decision and weighing the credibility of sources extend the ledger beyond this paper.

Ethics Statement

The decision streams use fictional entities and contain no personal data, and hosted models were queried on this synthetic data only. Systems that revise decisions on a user’s behalf can act on false or adversarial news; routing should be paired with

source verification and, for consequential decisions, with user confirmation.

References

- Souhail Bakkali and Ayoub Merimi. 2026. Amnesia by design, memory by necessity: Persistent state for document intelligence. *arXiv preprint arXiv:2609.32041*.
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*.
- Yize Cheng, Arshia Soltani Moakhar, Chenrui Fan, Parsa Hosseini, Kazem Faghieh, Zahra Sodagar, Wenxiao Wang, and Soheil Feizi. 2026. Your LLM agents are temporally blind: The misalignment between tool use decisions and human time perception. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 37082–37104.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12.
- Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.
- Pengfei Du. 2026. Memory for autonomous LLM agents: Mechanisms, evaluation, and emerging frontiers. *arXiv preprint arXiv:2603.07670*.
- Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2025. Test of time: A benchmark for evaluating LLMs on temporal reasoning. In *International Conference on Learning Representations*.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

- Xunyi Jiang, Dingyi Chang, Julian McAuley, and Xin Xu. 2026. When benchmarks age: Temporal misalignment through large language model factuality evaluation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 500–512.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What’s the answer right now? In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*.
- Nora Kassner, Øyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861.
- Yujin Kim, Jaehong Yoon, Seonghyeon Ye, Sangmin Bae, Namgyu Ho, Sung Ju Hwang, and Se-Young Yun. 2024. Carpe diem: On the evaluation of world knowledge in lifelong language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.
- Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liška, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d’Autume, Tomáš Kočiský, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. In *Advances in Neural Information Processing Systems*, volume 34.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33.
- Guanzhen Li, Liangming Pan, and Leye Wang. 2026. ProEvent: An event-centric benchmark for proactive agents. *arXiv preprint arXiv:2607.17701*.
- Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d’Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsonan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, Weiwen Liu, Yasheng Wang, Zhiyuan Liu, Fangming Liu, and Maosong Sun. 2025. Proactive agent: Shifting LLM agents from reactive responses to active assistance. In *International Conference on Learning Representations*.
- Yaxing Lyu, Shengjie Zhou, Binbin Toh, Pengyu Zhu, and Lijun Li. 2026. KC-Bench: A dynamic interactive benchmark for evaluating knowledge conflicts in LLM agents. *arXiv preprint arXiv:2609.03588*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35.
- Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. NoLiMa: Long-context evaluation beyond literal matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*.
- Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simula-lacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*.

- Qinglong Shi, Donghai Wang, Hantao Zhou, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, and Renqing He. 2026. Long-term task-oriented agent: Proactive long-term intent maintenance in dynamic environments. *arXiv preprint arXiv:2601.09382*.
- Yuanbo Tang, Huaze Tang, Tingyu Cao, Lam Nguyen, Anping Zhang, Xinwen Cao, Chunkang Liu, Wenbo Ding, and Yang Li. 2026. ProAgentBench: Evaluating LLM agents for proactive assistance with real-world data. *arXiv preprint arXiv:2602.04482*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*.
- Anxin Tian, Yiming Li, Xing Li, Hui-Ling Zhen, Lei Chen, Xianzhi Yu, Zhenhua Dong, and Mingxuan Yuan. 2026. SwiftMem: Fast agentic memory via query-aware indexing. *arXiv preprint arXiv:2601.08160*.
- Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*.
- Bryan Wilie, Samuel Cahyawijaya, Etsuko Ishii, Junxian He, and Pascale Fung. 2024. Belief revision: The adaptability of large language models reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10480–10496.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Xiaohongshu Dots Studio and Evolvent AI. 2026. ViBeLifeBench: Can your life agent be proactive and persistent in a living world? *arXiv preprint arXiv:2608.10875*.
- Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024a. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*.
- Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024b. TravelPlanner: A benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*.
- Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38.
- An Yang et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.
- Zhiyuan Zhai, Ming Li, and Xin Wang. 2026. Revisable by design: A theory of streaming LLM agent execution. *arXiv preprint arXiv:2604.23283*.
- Yikai Zhang, Siyu Yuan, Caiyu Hu, Kyle Richardson, Yanghua Xiao, and Jiangjie Chen. 2024. TimeArena: Shaping efficient multitasking language agents in a time-aware simulation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

A Formal Results

Proof of Proposition 1. Let T be the time from the last verification of f until f next changes. A constant hazard rate λ_f makes T exponential, $\Pr[T > \tau] = e^{-\lambda_f \tau}$. The fact holds at t_a exactly when $T > t_a - t_v = \Delta_{\text{fresh}} + \Delta_{\text{act}}$, which gives

Eq. 2. Since $\Delta_{\text{fresh}} \geq 0$, the first factor is at most one, so the probability is at most $e^{-\lambda_f \Delta_{\text{act}}}$, with equality exactly when the evidence is verified at the moment of decision. Fresher evidence moves the probability towards the ceiling; only information that arrives after t_d moves the decision itself. $\square$

Decisions with several dependencies. If the chosen option stays best only while each of the facts $f_1, \dots, f_n$ holds, and the facts change independently, the same argument gives

$$\Pr[\text{all } f_k \text{ hold at } t_a] \leq \exp \left(-\Delta_{\text{act}} \sum_{k=1}^n \lambda_{f_k} \right).$$

The ceiling tightens with every dependency, including dependencies the decision text leaves implicit, such as the substation behind a district. This is why the ledger records dependencies with the choice.

Derivation of Eq. 3. A decision is correct when the model keeps it if it should stand and switches it if it should change. Conditioning on the label and, for decisions that should change, on whether a decisive item was delivered, the law of total probability gives

$$\begin{aligned} A &= (1 - \pi) \Pr[\text{keep} \mid \text{stand}] \\ &\quad + \pi \rho \Pr[\text{switch} \mid \text{delivered}] \\ &\quad + \pi(1 - \rho) \Pr[\text{switch} \mid \text{missed}], \end{aligned}$$

which is Eq. 3. The identity holds for every router and model; the theory’s content lies in the predictions about its terms.

Comparing routers. Consider two routers used with the same model, with recalls ρ and ρ' and keep rates K and K' . If S and L are the same under both, Eq. 3 gives

$$A' - A = \pi(\rho' - \rho)(S - L) + (1 - \pi)(K' - K).$$

A router that reaches more decisive news gains accuracy in proportion to $S - L$, the margin by which the model acts on delivered news over undelivered news, and loses it through the misleading items that lower K . Strong judgment makes $S - L$ large and keeps K stable, so recall dominates; weak judgment shrinks $S - L$ and lowers K with every delivery, so exposure dominates. This is prediction P3 in closed form.

B Decision Streams

The generators, the three streams and their SHA-256 hashes are released with the paper; regenerating each stream from its seed reproduces the hash recorded with the model answers. Every label is computed from timestamps and stated facts, and every stream is checked by assertions before use.

DIRECT and STRESS. Decision times fall within the first three days of a six-day stream. Lags are drawn uniformly from 3–12 or 24–72 hours with equal probability, and action times fall in domain windows (dinner 19:00–22:00, trains 06:00–20:00, orders 08:00–20:00, permits on weekdays 09:00–15:30, hikes 07:00–13:00). Decisive items are published inside the revision window and block the option from before t_a until at least an hour after the activity; late items start at least one hour after the activity ends. Two DIRECT notices: “Notice from Juniper Barrel: closed for a private event from Tue 13 Oct 2026, 14:40 until Wed 14 Oct 2026, 01:10.” (names the option) and “City notice: planned power cut in the Riverside district from Wed 14 Oct 2026, 14:40 to Thu 15 Oct 2026, 05:10; restaurants and shops there will close.” (names the stated dependency). The generator asserts that $t_d < t_r < t_a$, that every decisive item lies in the revision window and overlaps the activity, that no item targets option 2, and that late items never overlap the activity. For STRESS, it recomputes all labels after placing decoys and paraphrasing notices, and asserts that no gold answer or exclusion changed and that no decoy relates to any decision.

INDIRECT. Each domain (walk-in dinners, rail journeys, day hikes, in-store pickups) has 16 bundles of a dependency, its other name, its upstream entity and four facilities. Each bundle serves one decision acting on days 2–3 and one on days 5–6, so the two are never open together; lags are 5–12 or 18–28 hours. A blocking notice for the activity starts between six hours and 15 minutes before it and ends at least 30 minutes after it; a late one starts at least an hour after the activity ends; a past one ends at least 45 minutes before it starts. Late, past and harmless notices use the four ways of reference equally, and blocking, late and past notices share templates, so only their times differ. After placing all notices, the generator recomputes every label and asserts the intended label, exactly two related notices per decision, the balance of the design, that decoys and unrelated notices concern no decision,

and that no notice affecting a decision was public before it was made.

B.1 A Decision from INDIRECT

Decision D0011 is the running example of Figure 1, reproduced verbatim from the stream. The dossier holds the other name of the district (3), the substation upstream of it (6), four facilities a walking diner does not need (2, 5, 10, 11), and six traps that mention other decisions’ entities (1, 4, 7, 8, 9, 12).

Times. Decided Tue 13 Oct 2026, 13:05; revisable until Tue 13 Oct 2026, 19:00; action Tue 13 Oct 2026, 20:30, lasting 2 hours. 30 notices are published in the revision window.

Request. The user wants to walk in for dinner at 20:30 on Tuesday 13 October 2026 (about 2 hours). They want the best-rated restaurant within a 20-minute walk and will walk there.

Decision-time facts. “Lovage Corner: rated 4.6/5; 9-minute walk from the user; in the Bellhaven district; open daily 12:00-02:00; walk-ins welcome.” “Red Bistro: rated 4.2/5; 8-minute walk from the user; in the Moorside district; open daily 12:00-02:00; walk-ins welcome.”

Choice. A. Lovage Corner (alternative: B. Red Bistro).

Dossier. (1) Tourist boats run from the Pigeon Lofts to the Bellhaven waterfront. (2) The Ironworks museum is in the Bellhaven district. (3) The Bellhaven district is known locally as the Chime Green. (4) The floodlights of the Bellhaven football ground are fed from the Dunsby substation. (5) Tram line T12 stops in the Bellhaven district. (6) The Bellhaven district is supplied with electricity by the Gorran substation. (7) The Bellhaven football club plays its derby against the Southbank district team. (8) Tram line T12 draws its power from the Hensall traction substation. (9) Tram line T12 continues from the Bellhaven district to the Millbrook district. (10) The Bellhaven district hosts the Bread market on weekday mornings. (11) The Bellhaven multi-storey car park serves visitors who drive. (12) Street musicians from the Leafy Triangle often play in the Bellhaven district.

Notices generated with this decision.

- Tue 13 Oct 2026, 14:50, related, harmless: “City notice: new street lighting installed in the Bellhaven district; no disruption expected.”
- Tue 13 Oct 2026, 15:05, same-name decoy: “Road closure: Bellhaven Road in the Ormsby district is shut from Tue 13 Oct 2026, 16:40 to Wed 14 Oct 2026, 01:15.”
- Tue 13 Oct 2026, 16:40, unrelated: “Notice from Tidewater Kitchen in the Ormsby district: closed from Tue 13 Oct 2026, 17:10 to Wed 14 Oct 2026, 01:40.”
- Tue 13 Oct 2026, 18:40, decisive (upstream entity): “Grid operator: emergency works at the Gorran substation from Tue 13 Oct 2026, 19:50 to Wed 14 Oct 2026, 00:15; every area it supplies will be without power and businesses there will close.”

Table 5 follows the decision through every router. The decision text and both dense routers miss the decisive notice, which names only the substation; BM25 over the whole dossier or over the watch conditions reaches it, and every model then switches. Under the decision text, Ministral 8B switches without the decisive notice, prompted by other news: the L term of Eq. 3.

Router	Delivered			Answer (gold: B)		
	Items	Decisive	Trap	Claude	Qwen	Ministral
None	0			A	A	A
Name match	0			A	A	A
BM25: decision	6			A	A	B
BM25: + dossier	5	✓	✓	B	B	B
Dense: + dossier	5			A	A	A
BM25: + watch cond.	8	✓	✓	B	B	B
Dense: watch cond.	11			A	A	A
Oracle: related	2	✓		B	B	B
Oracle: decisive	1	✓		B	B	B
All items	30	✓	✓	B	B	A

Table 5: Decision D0011 under every router: notices delivered, whether they include the decisive notice and a trap, and each model’s answer (bold: correct).

C Prompts

Each condition pairs a watch-condition prompt with a re-decision prompt. Every router of a condition uses the same re-decision prompt, so routers differ only in the notices they deliver.

DIRECT. Both prompts use the system message “*You are an assistant that manages decisions made on behalf of a user. Base your answers only on the information given.*” Watch conditions:

```
A decision has been made for the user and will be acted on
later. New information will keep arriving until the
revision deadline. Write the decision's watch
conditions: the kinds of new information that should
make us reconsider it.
Decision time: {t_d}
Evidence at decision time:
[F1] ...
Request: {request}
Options:
A. ...
B. ...
Decision made: {letter}. {option}
The user will act {t_a}. The decision can be changed until {
t_r}.
Write 3 to 6 watch conditions. Each must describe a specific
event that would make the chosen option unsuitable at
the action time. Cover events that affect the chosen
option directly, and events that affect it indirectly
through the districts, operators, platforms, regions or
services it depends on according to the evidence. Name
those entities explicitly.
Reply with JSON only, in this form:
{"watch": ["condition 1", "condition 2", "condition 3"]}
```

Re-decision:

```
Current time: {t_r}
Evidence at decision time ({t_d}):
[F1] ...
Request: {request}
Options:
A. ...
B. ...
Current decision: {letter}. {option}
New information published since the decision:
[N1] (published {t_w}) ...
The user will act {t_a}. This is the last chance to change
the decision. Taking the new information into account,
which option should the user go with? Ignore
information that is unrelated to these options or that
only takes effect after the user's activity is over.
Reply with JSON only: {"answer": "<letter>"}
```

STRESS. The system message is “*You revise previously made choices using ONLY the facts and*

notices provided. Respect the action time and notice effective time.” Claude writes watch conditions with:

```
A decision was made for a later action. Propose 3-6 specific
watch conditions that might force revisiting the chosen
option.
Use ONLY the decision-time evidence; name the chosen option
and its exact dependency entities if stated.
Prioritize notices affecting availability and execution,
including indirect disruptions. Avoid generic user-
payment, ratings or competing alternatives.
Decision made at: {t_d}
Last revision deadline: {t_r}
Action starts at: {t_a}
Request: {request}
Facts:
[F1] ...
Options:
A. ...
B. ...
Chosen: {letter}. {option}
Reply with JSON only: {"watch":["...", "...", "..."]}
```

Re-decision:

```
Current decision revision time: {t_r}
Decision-time facts ({t_d}):
[F1] ...
Request: {request}
Options:
A. ...
B. ...
Original choice: {letter}. {option}
Newly published notices (shown in time order; may be
unrelated or not yet effective):
[N1] (published {t_w}) ...
Action starts: {t_a}. Activity lasts {h} hours.
You must pick the option that will best satisfy the request
DURING the action interval, not when the notice was
published.
Ignore unrelated notices and changes whose effects start
after the activity ends. If the original choice remains
best, keep it.
Record only JSON {"answer":"A"} or {"answer":"B"} using the
required structured output protocol.
```

INDIRECT. The system message is “*You revise previously made choices using ONLY the facts and notices provided. Respect the action time and when each notice takes effect.*” Claude writes watch conditions with:

```
A decision was made for a user and will be acted on later.
New notices will keep arriving until the revision
deadline. Write the watch conditions for this decision:
the kinds of notice that should make us revisit it.
Decision made at: {t_d}
Revision deadline: {t_r}
Action starts: {t_a}; it lasts {h} hours.
Request: {request}
Decision-time facts:
[F1] ...
Options:
A. ...
B. ...
Chosen: {letter}. {option}
Background facts about the chosen option (from the ledger):
[B1] ...
Write 3 to 8 watch conditions. Each must describe a specific
event that would stop the user's activity with the
chosen option during the action. Name every entity the
activity depends on, including any other name it is
known by and anything upstream it relies on. Leave out
facilities and entities the user does not need for this
activity.
Reply with JSON only: {"watch":["...", "...", "..."]}
```

Re-decision:

```
Current time (last chance to revise): {t_r}
Decision-time facts ({t_d}):
[F1] ...
Request: {request}
Options:
A. ...
B. ...
Original choice: {letter}. {option}
Background facts about the original choice (from the decision
ledger):
[B1] ...
Newly published notices (in time order; they may be unrelated
or not in effect during the activity):
[N1] (published {t_w}) ...
Action starts: {t_a}. The activity lasts {h} hours.
Pick the option that will best satisfy the request DURING the
activity, not when the notice was published.
Ignore notices about other things and changes that are not in
effect during the activity. If the original choice
remains best, keep it.
Reply with JSON only: {"answer":"A"} or {"answer":"B"}.
```

D Implementation

Models and decoding. Qwen3-8B runs with thinking disabled through Hugging Face Transformers (Wolf et al., 2020) with greedy decoding, at most 400 output tokens for watch conditions and 128 for re-decisions. Claude Sonnet 4.6, GPT-6.1 Sol, Qwen3-235B-A22B (2507), Ministral 8B (Ministral 3 8B Instruct) and Mistral 7B (Instruct v0.2) run at temperature 0 with structured output and a plain-JSON fallback, with at most 768 output tokens per answer. Failed or invalid calls are retried and otherwise recorded as missing, never as an answer, and every answer is cached under a hash of the stream, model, router and prompt.

Routing. BM25 uses $k_1=1.5$ and $b=0.75$ over lowercase alphanumeric tokens; dense routing uses BGE-base v1.5 (Xiao et al., 2024). Dependency queries use “{option} unavailable cancelled blocked closed or delayed”, “{dependency} outage closure disruption maintenance restriction strike suspended” and “{dependency} operating hours availability travel safety affects {option}”. The time filter parses a start time introduced by *from*, *starting*, *beginning*, *effective* or *start* and drops the decision when that time is not before the end of its activity.

Analysis. Every number in the paper is computed by scripts from the per-decision logs, which record for each decision, router and model the notices delivered, whether the decisive notice and traps were among them, and the answer. The scripts and logs are released with the paper.

E Additional Results

Where the errors are. Table 6 breaks down INDIRECT accuracy by how the decisive notice refers

	Name	BM25	+Doss.	+Watch	Oracle	All
<i>Needed switches made, by how the notice refers to the option</i>						
option name	100.0	100.0	100.0	100.0	100.0	100.0
dependency	8.3	100.0	100.0	97.9	100.0	97.9
other name	6.2	54.2	95.8	97.9	97.9	93.8
upstream entity	4.2	35.4	93.8	75.0	100.0	89.6
<i>Decisions kept, by the related news they received</i>						
starts after	81.2	52.1	37.5	43.8	43.8	39.6
already over	77.1	37.5	27.1	33.3	33.3	27.1
harmless	91.7	68.8	52.1	70.8	72.9	52.1
not needed	89.6	70.8	52.1	60.4	62.5	43.8

Table 6: Needed switches made and decisions kept on INDIRECT, by how the decisive notice refers to the option and by the related news a decision received (% , mean over the three models with complete answers; 16 decisions per row and model). Routing sets which switches are possible; judgment sets the rest.

to the option and by the related news a decision received. Routing sets which switches are possible: the decision text reaches the option and its stated dependency, and the dossier and watch conditions extend the reach to other names and upstream entities. Judgment sets the rest: decisions with a change that is already over or starts after the activity are the ones most often abandoned.

Statistical tests. Table 7 lists the exact McNemar tests behind the INDIRECT comparisons in Section 7.

Every router on DIRECT. Table 8 reports every router with Qwen3-8B on DIRECT, and Table 9 breaks down its needless switches by the related news a decision received. With oracle routing, Qwen3-8B makes every needed switch and keeps 62.0% of the decisions that should stand. BM25 over the decision text, scored on DIRECT for routing only, delivers 97.1% of decisive pairs.

Routing with other K . Table 10 shows INDIRECT routing as the number K of decisions per notice varies. With $K=1$, the whole dossier delivers 85.9% of decisive notices and watch conditions 71.9%.

A second STRESS sample. Table 11 shows Claude Sonnet 4.6 on a second sample of 73 decisions, where routing again lifts accuracy from 61.6% without routing to 89.0% with BM25, against 94.5% for the related oracle.

Mistral 7B on INDIRECT. Mistral 7B returned a valid answer for 74–95% of the prompts, depending on the router, and mostly kept its original choice: given only decisive notices, it switched on 48.9% of the decisions it answered. The comparisons

in Section 7 use the three models with complete answers.

Router A	Router B	Claude		Qwen-235B		Ministral 8B		Pooled	
		W-L	<i>p</i>	W-L	<i>p</i>	W-L	<i>p</i>	W-L	<i>p</i>
BM25: decision	Name match	26-5	<.001	30-26	.689	30-26	.689	86-57	.019
BM25: + dossier	BM25: decision	22-4	<.001	17-15	.860	12-13	1.000	51-32	.048
BM25: + watch cond.	BM25: decision	21-5	.002	16-11	.442	15-7	.134	52-23	.001
BM25: + watch cond.	BM25: + dossier	5-7	.774	9-6	.607	12-3	.035	26-16	.164
Dense: watch cond.	Dense: + dossier	11-13	.839	12-16	.572	10-9	1.000	33-38	.635
All items	BM25: + dossier	6-8	.791	2-9	.065	10-8	.815	18-25	.360
All items	BM25: + watch cond.	6-6	1.000	6-16	.052	8-15	.210	20-37	.033
Oracle: related	BM25: + dossier	9-1	.021	11-0	<.001	11-5	.210	31-6	<.001
Oracle: decisive	Oracle: related	1-0	1.000	38-0	<.001	52-0	<.001	91-0	<.001

Table 7: Exact McNemar tests on INDIRECT. W-L: decisions router A gets right and router B gets wrong, and the reverse; 128 decisions per model, pooled over the three models with complete answers. *p*-values are two-sided and uncorrected.

Router	Routing (%)			Accuracy	Final decisions (%)			Tokens (k)	<i>p</i>
	Recall	Implicit	Precision		Switch: named	Switch: dependency	Keep		
<i>Qwen/Qwen3-8B</i>									
None (freshness at t_d)	0.0	0.0	–	45.5	0.0	0.0	100.0	0.0	<0.001
Name match	39.7	0.0	100.0	70.0	100.0	18.2	88.0	23.9	0.851
Dense similarity	83.8	75.6	14.3	71.8	92.0	63.6	66.0	102.4	–
Watch conditions	77.9	65.9	14.2	62.7	80.0	57.6	56.0	146.9	0.076
Oracle: related	100.0	100.0	100.0	82.7	100.0	100.0	62.0	44.2	0.050
All items	100.0	100.0	1.6	55.5	60.0	33.3	68.0	803.7	0.003

Table 8: Qwen3-8B on DIRECT. *Recall*: share of decision-changing items delivered to the decision they change; *Implicit*: the same for items that reach the option only through a dependency; *Precision*: share of delivered items that concern the chosen option. *Switch: named / dependency*: share of decisions that should switch and did, when the evidence names the option or reaches it only through a dependency; *Keep*: share of decisions that should stand and did. *Tokens*: prompt tokens spent on routing and re-deciding, including writing watch conditions. *p*: exact McNemar test on per-decision correctness against the most accurate practical router (bold; oracle excluded). Routing recall and precision for None, Name match, Oracle and All follow from the stream and are identical across models.

Related news	<i>n</i>	Oracle	Name	Sim.	Watch	All
Late change	18	94	28	39	67	33
Harmless only	19	11	5	37	37	32
None	13	0	0	23	23	31

Table 9: Needless switches (%) on decisions that should stand, by the related news published in their revision window. *Late change*: a change to the option or its dependency that starts only after the activity ends; *Harmless only*: news about the option or its dependency that does not affect it. Name: name match; Sim.: similarity to the decision; Watch: watch conditions.

Router	<i>K</i> =1		<i>K</i> =3		<i>K</i> =5	
	Recall	Traps	Recall	Traps	Recall	Traps
BM25: decision	51.6	4	62.5	17	68.8	35
BM25: + dossier	85.9	21	100.0	89	100.0	96
Dense: + dossier	70.3	11	85.9	59	90.6	74
BM25: + watch cond.	71.9	5	92.2	20	96.9	38
Dense: watch cond.	75.0	4	84.4	27	90.6	53

Table 10: INDIRECT routing as the number *K* of decisions each notice is sent to varies: decisive pairs delivered (%) and lexical traps delivered.

Router	Accuracy	Keep	Needless
None	61.6	100.0	0
Name match	75.3	91.1	4
BM25: decision	89.0	91.1	4
BM25: + time filter	91.8	93.3	3
Dense: decision	91.8	93.3	3
Dense: watch cond.	89.0	86.7	6
Dependency queries	90.4	88.9	5
Oracle: related	94.5	93.3	3
Oracle: decisive	100.0	100.0	0

Table 11: Claude Sonnet 4.6 on a second STRESS sample (73 decisions). *Keep*: accuracy on decisions that should stand (%); *Needless*: switches on those decisions. The table lists the routers run on this sample.