SPARSE CONTEXT SYNTHESIS: A FORMAL FRAMEWORK FOR OFF-THE-SHELF LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Context optimization improves off-the-shelf large language model (LLM) agents by dynamically injecting context into their prompts rather than fine-tuning model weights. We study a common paradigm of this approach: a catalog of gated *factoids*, which are small, independently authored functions that inspect an episode and emit short, actionable claims, such as relevant files or policy constraints. We formalize this process as *sparse context synthesis*: for a given episode, a sparse sub-catalog is activated, and its claims are embedded, weighted on the simplex, and synthesized by the agent into a free-text response. Within this framework, we make three primary contributions. First, we measure how well a catalog fits the evidence by the mutual information its outputs share with a latent target, formalizing three competing design pressures: coverage, redundancy, and localization. Notably, we demonstrate that negative redundancy signals structural synergy among factoids. Second, we establish a theoretical error bridge connecting catalog omissions to downstream agent error, proving that the distance between the embeddings of complete versus omitted claim sets is tightly bounded by their precision and recall deficits. We instantiate this for both observation claims and causal claim graphs to bound end-to-end omission errors. Third, we theoretically ground context importance weights by relating them to expectation-maximization (EM) responsibilities, the Barber–Agakov bound, and information projections, delineating formal proofs from structural analogies. Finally, numerical evaluations on SWE-bench Verified and τ^2-bench illustrate the pipeline's operational dynamics, revealing when injected claims successfully bind to guide agent behavior.

1 INTRODUCTION

As large language models (LLMs) increasingly serve as the core reasoning engines for autonomous agents, performance improvements have expanded beyond traditional model weight updates. Agentic harness optimization has emerged as a crucial paradigm, shifting the focus toward algorithmically searching and evolving the code, tool interfaces, and control loops surrounding an LLM to maximize task efficacy (Meng et al., 2026). While harness engineering encompasses a broad range of optimization verticals, this work focuses specifically on agentic context adaptation (ACA). In ACA, system prompts, contextual state, and tool definitions are treated as dynamic playbooks that update iteratively through agent reflection and performance feedback (e.g., Zhang et al., 2026). To advance the state of the art in ACA, we draw inspiration from Loom (Begleiter et al., 2026), a recently proposed framework for debugging failures in complex distributed compute systems, originally designed for Root Cause Analysis (RCA).

Related Work. Our work bridges agentic context adaptation (Shinn et al., 2023; Zhang et al., 2026) and information-theoretic context bounds (Cover & Thomas, 2006; Xu et al., 2020; Colaco & Lahjouji, 2026). By viewing gated claims through the lenses of weak supervision (Ratner et al., 2016), specialists in online learning (Freund et al., 1997), and retrieval and long-context limitations (Lewis et al., 2020; Liu et al., 2024), we formalize what catalog-generated context reveals about the episode's latent target. Unlike prior theory, which analyzes the harness search loop (Shi et al., 2024; Cai et al., 2026) or scores and selects context from a given pool (Mei et al., 2025; Kim, 2026), our

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

analysis specifically targets gated, episode-specific claim catalogs. See Appendix A for an extended literature review.

Contributions. We build on Loom's catalog-and-consensus design and turn it into a general framework for agentic harness optimization, making four key contributions. First, we develop an *information-theoretic framework* that scores context catalogs by bounding mutual information using the Barber–Agakov bound, interpreting the consensus weights through EM responsibilities and m-projections. Second, we formalize *catalog design pressures* (coverage, redundancy, localization), showing that redundancy is negative when factoids are informative only jointly (synergy), to guide Agentic Context Adaptation (ACA). Third, we establish *representation error bounds*: the embedding distance between claim sets is tightly bounded by their precision and recall, which bounds the error caused by catalog omissions. Finally, we provide a *domain-agnostic generalization*, adapting this design into an ACA pipeline. Evaluations in software engineering and customer support highlight the operational complexities of context injection: while injected claims improve file localization and policy adherence, they do not uniformly increase final task success rates.

2 Notation and Standing Definitions

This section fixes the random variables, the generative model, and the variational stand-ins used throughout. Later sections instantiate these objects; they do not change their meaning.

2.1 Observables, latents, and the catalog

An episode is an observed context X, taking values in a space $\mathcal{X}$: everything the agent can inspect, such as a code repository and an issue, a customer conversation and a database, or system logs. The unobserved target is a discrete latent $Z \in \mathcal{Z} = \{1, \dots, K\}$: the quantity the agent must infer to succeed, such as the files that need editing, the policy-relevant record, or the root cause of a failure. A fixed catalog $\mathcal{F} = \{f_1, \dots, f_M\}$ of template language functionals, which we refer to as *factoids* (or *channels*), maps each x to texts. Indices run over $[M] = \{1, \dots, M\}$, and the *active index set* for an episode x is $\mathcal{A}(x) \subseteq [M]$, so that sums such as $\sum_{i \in \mathcal{A}(x)}$ range over indices rather than over functions. Write $S_{\mathcal{F}}(X)$ for the (random) factoid outputs induced by $\mathcal{F}$ on X. Consensus produces a simplex weight $w(x)$ over the active factoids; the agent then emits a free-text output $\hat{y}$.

Distributions: true versus model. Unsubscripted quantities, P_X (the marginal law of X), $H(Z)$, $I(Z; X)$, $U(\mathcal{F})$, $\Gamma(\mathcal{F})$, are taken under the true joint law P of (X, Z); a subscript θ marks the same quantity under the model p_θ introduced below. The two coincide only when the model is correct.

2.2 The generative model θ

The parameter $\theta \in \Theta$ indexes a joint factorization

$$p_\theta(X, Z) = p_\theta(Z)\, p_\theta(X \mid Z)\,: \tag{1}$$

a prior over targets and a likelihood of observations given a target. Every quantity with a θ subscript, $p_\theta(Z \mid X)$, $H_\theta(Z \mid X)$, $I_\theta(X; Z)$, is computed *inside this model*, not in the unknown true world. Fitting θ is the M-step of EM (Appendix C): change the target–observation model so the observed data become more likely. We do not commit to a concrete parametric family (mixture, HMM, neural likelihood); θ is a placeholder for whatever generative model of the task one would run EM on.

2.3 The variational stand-in q

The model posterior $p_\theta(Z \mid X)$ involves a sum over the K values of Z and is typically intractable. A variational distribution $q(Z \mid X)$ is any distribution over targets, given the observation, that we *can* compute and that we use in place of $p_\theta(Z \mid X)$. It is a computational knob, not a parameter of nature.

Three related stand-ins appear below:

1. $q(Z \mid X)$: variational posterior over target values (the EM / Barber–Agakov object).

2. $\hat{q}(Z \mid S_{\mathcal{F}})$: a trained classifier from factoid texts to targets, used to estimate catalog utility (Appendix E).
3. $\beta(J \mid X)$: factoid weights $w(x)$ read as the conditional law of a random factoid index J taking values in $\mathcal{A}(X)$, not a distribution over Z (Section 4).

In the consensus reweighting of Section 4, the E-step equality $q(Z \mid X) = p_\theta(Z \mid X)$ is often unavailable, and β is only a guess over which factoid to trust. Appendix B tabulates θ against q and states the entropy and mutual-information conventions used below, including what zero conditional entropy does and does not imply.

3 Mathematical Mapping and Architecture

Given an episode $x \in \mathcal{X}$ (notation as in Section 2), the system architecture is decomposed into the following sequential operations:

Dynamic Sub-Dictionary Generation. Let $\mathcal{F} = \{f_1, \ldots, f_M\}$ be a fixed, global catalog of template language functionals. For a specific input x, a subset of these functionals is activated; their indices form the active index set

$$\mathcal{A}(x) \subseteq [M]. \tag{2}$$

Each active functional emits a dynamic text $t_i = f_i(x)$, $i \in \mathcal{A}(x)$.

Semantic Vector Embedding. Let $\mathcal{T}$ denote the space of texts. An embedding function $\mathrm{emb} : \mathcal{T} \to \mathbb{R}^d$ maps these emitted texts into a continuous semantic vector space. The active sub-dictionary for input x is formalized as:

$$D(x) = \{v_i(x) = \mathrm{emb}(f_i(x)) \mid i \in \mathcal{A}(x)\} \subset \mathbb{R}^d \tag{3}$$

Simplex Weighting and Semantic Prototype. The system learns an importance weighting vector $w(x)$ over the active components. This weighting defines a convex combination (a semantic centroid) over the active sub-dictionary:

$$\bar{v}(x) = \sum_{i \in \mathcal{A}(x)} w_i(x) v_i(x), \quad \text{s.t.} \quad w_i \geq 0, \sum w_i = 1 \tag{4}$$

Here, $\bar{v}(x)$ represents a point inside the convex hull of the active embeddings, $\mathrm{conv}(D(x))$. The probabilistic mechanism that produces $w(x)$ is developed in Section 4.

Non-Linear Generative Synthesis (Decoder). The agent, an off-the-shelf LLM, acts as a non-linear decoding operator Ψ, which maps the weighted semantic representation back onto the natural language manifold to produce the final free-text output $\hat{y}$:

$$\hat{y} = \Psi(\mathcal{A}(x), w(x)) \tag{5}$$

The reference output against which $\hat{y}$ is evaluated is a free text denoted $y^*(x)$.

4 Consensus Weights as a Variational Surrogate

While Section 3 defines the semantic centroid $\bar{v}(x)$ in (4) over the active simplex $\Delta^{|\mathcal{A}(x)|-1}$, we now give a probabilistic reading of the importance weighting vector $w(x)$. In practice, $w(x)$ is produced by iterative embedding-centroid reweighting, which we interpret as a factoid-level variational surrogate; we also relate it to classical EM responsibilities over latent targets (Dempster et al., 1977; Bishop, 2006), and connect it to the InfoMax principle (Barber & Agakov, 2003) and information geometry (Amari & Nagaoka, 2000). Appendix C restates the generative setup, discusses the size of K, and recalls the ELBO and the EM algorithm.

4.1 Consensus Reweighting and Its Variational Reading

Begleiter et al. (2026) compute consensus weights by iterative embedding-centroid reweighting, which we interpret here as a *factoid-level variational surrogate*.

Reweighting procedure. Let $\{v_i(x)\}_{i\in\mathcal{A}(x)}$ be the active embeddings and $w^{(0)}$ an initialization (e.g., expert reliability priors, optionally adjusted for static redundancy groups). The procedure alternates:

$$c^{(t)} = \frac{\sum_{i\in\mathcal{A}(x)} \tilde{w}_i^{(t)}\, v_i(x)}{\sum_{i\in\mathcal{A}(x)} \tilde{w}_i^{(t)}}, \tag{6}$$

$$w_i^{(t+1)} \propto \mathrm{sim}\big(v_i(x),\, c^{(t)}\big), \tag{7}$$

where $\tilde{w}^{(t)}$ denotes redundancy-adjusted weights and sim is cosine similarity (mapped to nonnegative scores and normalized so that $w^{(t+1)} \in \Delta^{|\mathcal{A}(x)|-1}$). At convergence one obtains $w(x)$ and the learned semantic prototype $\bar{v}_{\mathrm{learned}}(x) = c^*$ used in Section 6.

Variational reading. After normalization, $w(x)$ is a categorical distribution on the active factoid index set:

$$\beta(J = i \mid X = x) \;:=\; w_i(x), \qquad i \in \mathcal{A}(x). \tag{8}$$

Thus $\beta(\cdot \mid X)$ is a valid variational distribution, but over *channels/hypotheses*, not over target values $Z \in \mathcal{Z}$. The centroid $c^{(t)}$ plays the role of a continuous variational summary of the consensus output in embedding space; the weight update is an EM-like responsibility step that reassigns mass to factoids by agreement with that summary. In this sense, the fixed point of this procedure is a tractable variational surrogate for factoid consensus when an explicit posterior $q(Z \mid X)$ is unavailable or when K is too large to materialize.

4.2 Certifying Uncertainty Reduction: The Barber–Agakov Bound

We relate these consensus weights to two classical viewpoints. Neither derives the reweighting update (7); both clarify what the weights are meant to approximate. The information-geometric view is developed in Appendix D.

InfoMax principle (Barber-Agakov bound). An informative representation should maximize the mutual information $I_\theta(X; Z)$ between observations X and latent targets Z:

$$I_\theta(X; Z) = H_\theta(Z) - H_\theta(Z \mid X) = \mathbb{E}_{p_\theta(X,Z)}\left[\log \frac{p_\theta(Z \mid X)}{p_\theta(Z)}\right]. \tag{9}$$

By the Barber-Agakov variational inequality (Barber & Agakov, 2003), for any variational posterior $q(Z \mid X)$,

$$I_\theta(X; Z) \geq \mathbb{E}_{p_\theta(X,Z)}\left[\log q(Z \mid X)\right] + H_\theta(Z) \equiv \mathcal{L}_{\mathrm{BA}}(q, \theta), \tag{10}$$

with equality if and only if $q(Z \mid X) = p_\theta(Z \mid X)$. Intuitively, $I_\theta(X; Z)$ measures how much uncertainty about the target is eliminated once the observation is read, and (10) certifies part of it as a difference of two uncertainties. The first, $H_\theta(Z)$, is the prior uncertainty of Appendix B: the baseline confusion about the target before looking at any data. The second, $-\mathbb{E}[\log q(Z \mid X)]$, is the average surprise of a computable predictor q, used in place of the intractable posterior, when the true target is revealed; equivalently, $\mathbb{E}[\log q(Z \mid X)]$ is the average log-probability that q assigns to the correct target after reading the observation. Appendix C discusses the bound's extreme cases and its relation to the EM steps.

5 Catalog-Evidence Fit

Having defined catalog activation and consensus weighting, we ask a design question that precedes end-to-end error accounting: how well does a fixed factoid catalog $\mathcal{F}$ fit the statistical nature of the evidence X? We take information theory as the primary language and statistical geometry as the geometric counterpart already used by factoid consensus (Appendix D).

5.1 Formal utility

Let $S_{\mathcal{F}}(X)$ denote the (random) factoid outputs induced by $\mathcal{F}$ on X, i.e. the active texts $\{f_i(X) : i \in \mathcal{A}(X)\}$ or their embeddings $D(X)$. Ideal catalog utility is the mutual information extracted

about the latent target:

$$U(\mathcal{F}) \;:=\; I\big(Z;\, S_{\mathcal{F}}(X)\big) \;=\; I(Z;X) - I\big(Z;\, X \mid S_{\mathcal{F}}(X)\big). \tag{11}$$

The second equality assumes that the factoid outputs depend on the target only through the observation, i.e. that $Z \to X \to S_{\mathcal{F}}(X)$ is a Markov chain. This is automatic when each f_i is a deterministic function of x, as for template factoids; for stochastic factoids (e.g. LLM-generated text) it asserts that their randomness is independent of Z given X. Under it, $S_{\mathcal{F}}(X)$ adds nothing about Z once X is known, so $I(Z; X, S_{\mathcal{F}}(X)) = I(Z;X)$, and the chain rule $I(Z; X, S_{\mathcal{F}}(X)) = I(Z; S_{\mathcal{F}}(X)) + I(Z; X \mid S_{\mathcal{F}}(X))$ gives (11). It also gives $U(\mathcal{F}) \le I(Z;X)$: a catalog cannot extract more information about the target than the observation contains.

Equivalently, $U(\mathcal{F}) = H(Z) - H\big(Z \mid S_{\mathcal{F}}(X)\big)$. A catalog *fits* X when $S_{\mathcal{F}}(X)$ nearly saturates the information that X carries about Z, i.e. when the residual $I(Z; X \mid S_{\mathcal{F}}(X))$ is small.

Per-episode quantities. Mutual information is an average over episodes and cannot be localized by conditioning on $X = x$: once the observation is known, the factoid outputs carry no further information about Z, so $I\big(Z; S_{\mathcal{F}}(X) \mid X = x\big) = 0$ for every x. Per-episode quantities are pointwise divergences instead. For deterministic factoids, write $s(x)$ for the outputs of the active factoids $\{f_i(x)\}_{i \in \mathcal{A}(x)}$ and define

$$\begin{aligned} U_x(\mathcal{F}) &:= D_{\mathrm{KL}}\big(P(Z \mid S_{\mathcal{F}} = s(x)) \,\big\|\, P(Z)\big), \\ \gamma(x) &:= D_{\mathrm{KL}}\big(P(Z \mid X = x) \,\big\|\, P(Z \mid S_{\mathcal{F}} = s(x))\big). \end{aligned} \tag{12}$$

Here U_x measures how far the factoid outputs of episode x move belief about the target away from the prior, and $\gamma(x)$ measures the information about Z in the observation x that its active factoids miss; $\gamma(x) = 0$ exactly when they are as informative as the full observation. Both are nonnegative, and averaging over episodes recovers the global quantities:

$$\mathbb{E}_{x \sim P_X}\big[U_x(\mathcal{F})\big] = U(\mathcal{F}), \qquad \mathbb{E}_{x \sim P_X}\big[\gamma(x)\big] = I\big(Z; X \mid S_{\mathcal{F}}(X)\big) = \Gamma(\mathcal{F}), \tag{13}$$

with $\Gamma(\mathcal{F})$ the coverage gap of (14). Since $s(x)$ involves only the factoids in $\mathcal{A}(x)$, this matches the large-K regime discussed in Appendix C: global cardinality may be huge, but fit is judged on the activated support.

5.2 Design pressures: coverage, redundancy, localization

Write $S_i = f_i(X)$ for factoid channels. Three diagnostics capture distinct design goals. They are not a partition of $U(\mathcal{F})$: only the coverage gap enters an exact identity, $I(Z;X) = U(\mathcal{F}) + \Gamma(\mathcal{F})$, and localization is a cardinality rather than an information quantity.

$$\text{Coverage gap} \quad \Gamma(\mathcal{F}) := I(Z;X) - U(\mathcal{F}) = I\big(Z; X \mid S_{\mathcal{F}}(X)\big), \tag{14}$$

$$\text{Redundancy} \quad \mathrm{Red}(\mathcal{F}) := \sum_{i \in [M]} I(Z; S_i) - I\big(Z; \{S_i\}_{i \in [M]}\big), \tag{15}$$

$$\text{Localization} \quad \mathrm{Loc}(\mathcal{F}) := \mathbb{E}_x\big[|\mathcal{A}(x)|\big] \quad \text{and} \quad \mathbb{E}_x\big[K_x\big], \tag{16}$$

where K_x is the effective posterior support size on $\mathcal{Z}$ given x (Appendix C). Small Γ means the catalog is informationally sufficient for Z, and moderate Loc means activation stays sparse.

The redundancy term has a sign. If two factoids duplicate each other, $S_1 = S_2$, their individual contributions count the same information twice and $\mathrm{Red}(\mathcal{F}) > 0$. If they are informative only jointly, $\mathrm{Red}(\mathcal{F}) < 0$. The simplest case is XOR: let S_1, S_2 be independent fair bits and $Z = S_1 \oplus S_2$, so $Z = 1$ exactly when the two bits differ. Knowing S_1 alone leaves Z a fair coin, and likewise for S_2, so $I(Z;S_1) = I(Z;S_2) = 0$; knowing both determines Z, so $I(Z; S_1, S_2) = 1$ bit, and $\mathrm{Red}(\mathcal{F}) = 0 + 0 - 1 = -1$ bit. The information about Z lies in the relationship between the two factoids, not in either one. Negative redundancy is therefore *synergy*: the catalog is worth more than the sum of its factoids. Because $\mathrm{Red}(\mathcal{F})$ is net redundancy minus synergy, a value near zero can hide large amounts of both. Consensus weighting down-weights outlying factoids; duplicates are discounted only through the redundancy-adjusted weights $\tilde{w}$ in (6). Appendix E gives finite-sample estimators of these quantities.

Design rule. Improve $\mathcal{F}$ by reducing Γ without inflating Red or Loc: add factoids that cut residual uncertainty about Z on mass of P_X, prune factoids that contribute only redundant mutual information, and keep activation local via $\mathcal{A}(x)$. Redundancy must be judged jointly: a factoid that is uninformative on its own may be synergistic with others, as in the XOR example of Section 5.2, and pruning it would lose information. Consensus weighting then operates on an already well-fitted active set. In practice, treat catalog edits (adding or removing factoids, changing gates) as regressions: accept a change only if $\hat{\Gamma}$ and the claim deficits of Section 6 do not worsen on a held-out slice, and $\widehat{\mathrm{Red}}$ and the mean active-set size stay within budget.

6 End-to-End Error Decomposition: Vector vs. Claim-Space

Given a catalog whose fit to X is characterized in Section 5, we next evaluate end-to-end pipeline error. Continuous vector distance ($\|\mathrm{emb}(y^*) - \mathrm{emb}(\hat{y})\|_2$) provides a geometrically elegant upper bound, but suffers from "embedding blur": failing to distinguish between factual omissions and causal inversions. To resolve this, we present a dual evaluation suite: a continuous vector-space bound and a discrete, claim-level causal path evaluation.

Scope. This section treats evaluation as set retrieval: each predicted object (a claim, or a unit of the causal graph) either matches a gold object or does not, and error is summarized by recall and precision over the matched sets. It does not model partial credit for nearly correct claims, the relative severity of different errors, or calibration and abstention, and it takes the matching procedure as given (in practice, an evaluation oracle or a judge). The causal layer stays within this framing: a graph is scored by matching over a chosen unit, so structure beyond that unit is captured only when the unit is a whole path.

6.1 Continuous Error Decomposition (Vector-Space)

Let $\bar{v}^*$ be the *oracle centroid*, the point of the active convex hull closest to the reference embedding,

$$\bar{v}^* := \operatorname*{arg\,min}_{v \in \mathrm{conv}(D(x))} \|\mathrm{emb}(y^*) - v\|_2, \tag{17}$$

and let $\bar{v}_{\mathrm{learned}}$ be the centroid formed by the consensus weights of (6)–(7). Writing $\text{Total Error}_{L_2} := \|\mathrm{emb}(y^*) - \mathrm{emb}(\hat{y})\|_2$ and applying the triangle inequality along the path $\mathrm{emb}(y^*) \to \bar{v}^* \to \bar{v}_{\mathrm{learned}} \to \mathrm{emb}(\hat{y})$ gives

$$\text{Total Error}_{L_2} \le \underbrace{\|\mathrm{emb}(y^*) - \bar{v}^*\|_2}_{\text{Catalog Omission Error}} + \underbrace{\|\bar{v}^* - \bar{v}_{\text{learned}}\|_2}_{\text{Weight Estimation Error}} + \underbrace{\|\bar{v}_{\text{learned}} - \mathrm{emb}(\hat{y})\|_2}_{\text{Agent Realization Error}} \tag{18}$$

This is a bound along a proxy path, not an orthogonal decomposition: the three stage terms can compound or cancel. The agent never forms $\bar{v}_{\mathrm{learned}}$ (it consumes $\mathcal{A}(x)$ and $w(x)$); the path is inserted for analysis. By (17), the first term is the distance from the reference embedding to the active convex hull, a property of the catalog alone.

6.2 Claim-Level Causal Path Evaluation (Discrete-Space)

We map the reference output y^* into a gold set of atomic claims $\mathrm{Gold}(y^*)$, and the synthesized text $\hat{y}$ into a generated claim set $\mathrm{Pred}(\hat{y})$. Claims are strictly categorized into *observations* (extracted factual states) and *conclusions* (the inferred target and its consequences).

Observation fidelity is evaluated as ordinary set matching over atomic claims. Causal fidelity is richer: conclusions induce a *causal claim graph* $G = (V, R)$, whose nodes V are (a subset of) claims and whose directed relations R encode causal support or explanation edges among them. Accordingly, we distinguish:

- **Discrete claim metrics** over observation sets: OBS-Recall, OBS-Precision.
- **Graphical causal metrics** over a fixed matching unit on G (typically directed edges, attributed relations, or causal paths): RC-Recall, RC-Precision.

Both precisions are needed because the two layers fail independently. Suppose the reference output states that a flight was cancelled and therefore the booking was refunded, and the prediction states that the booking was refunded and therefore the flight was cancelled. The observation claims coincide (the flight was cancelled; the booking was refunded), so OBS-Precision = OBS-Recall = 1. The single causal edge is reversed, so with directed edges as the matching unit RC-Precision = RC-Recall = 0. Because the two outputs use nearly the same words, their embeddings can be close: this is the causal inversion that a vector distance blurs and the graph layer detects. Conversely, a prediction can recover the correct causal chain while adding a spurious observation, which lowers OBS-Precision but not RC-Precision.

An evaluation oracle validates both layers against the recorded episode. This yields a layer-localized discrepancy triple, whose coordinates are paired with the three vector-space stages in the pipeline localization below:

$$\text{Total Discrepancy} = \Big(\underbrace{1 - \text{Recall}_{\mathcal{F}}}_{\text{Catalog Coverage Deficit}}, \underbrace{1 - \text{OBS-Precision}}_{\text{Fact Extraction Error}}, \underbrace{1 - \text{RC-Precision}}_{\text{Causal Graph Error}} \Big) \tag{19}$$

6.3 Mathematical Bridge: A Generic Error Bound over Embeddable Objects

The continuous geometric norm and the discrete evaluation metrics are linked by the same algebraic argument, applied at two semantic layers. We first state the machinery abstractly, then instantiate it for atomic claims and for causal graphs.

Generic setting. Let Ω be a domain of embeddable evaluative objects, and let $\mathrm{emb}_\Omega : \Omega \to \mathbb{R}^d$ map each object into a semantic vector space with uniformly bounded norms $\|\mathrm{emb}_\Omega(\omega)\|_2 \le B$. Given a gold collection $\mathrm{Gold}_\Omega \subseteq \Omega$ and a predicted collection $\mathrm{Pred}_\Omega \subseteq \Omega$, define the set embeddings as compositional averages

$$\begin{aligned} \mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) &:= \frac{1}{|\,\mathrm{Gold}_\Omega\,|} \sum_{\omega \in \mathrm{Gold}_\Omega} \mathrm{emb}_\Omega(\omega), \\ \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega) &:= \frac{1}{|\,\mathrm{Pred}_\Omega\,|} \sum_{\omega' \in \mathrm{Pred}_\Omega} \mathrm{emb}_\Omega(\omega'). \end{aligned} \tag{20}$$

and partition into true, missed, and false objects:

$$\Omega_{\mathrm{true}} := \mathrm{Pred}_\Omega \cap \mathrm{Gold}_\Omega, \quad \Omega_{\mathrm{missed}} := \mathrm{Gold}_\Omega \setminus \mathrm{Pred}_\Omega, \quad \Omega_{\mathrm{false}} := \mathrm{Pred}_\Omega \setminus \mathrm{Gold}_\Omega\,.$$

Theorem 1 (Error bridge)**.** *Let Ω be any domain of embeddable objects as above, and let Gold_Ω and Pred_Ω be finite nonempty subsets of Ω, with*

$$\mathrm{Recall}_\Omega := \frac{|\Omega_{\mathrm{true}}|}{|\,\mathrm{Gold}_\Omega\,|}, \qquad \mathrm{Precision}_\Omega := \frac{|\Omega_{\mathrm{true}}|}{|\,\mathrm{Pred}_\Omega\,|}.$$

Then

$$\begin{aligned} \|\mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) - \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega)\|_2 &\le B\Big(\big|\mathrm{Recall}_\Omega - \mathrm{Precision}_\Omega\big| \\ &\qquad + (1 - \mathrm{Recall}_\Omega) + (1 - \mathrm{Precision}_\Omega) \Big) \\ &= 2B \max\big(1 - \mathrm{Recall}_\Omega,\ 1 - \mathrm{Precision}_\Omega\big). \end{aligned} \tag{21}$$

The bound is attained: for $\mathrm{Gold}_\Omega = \{a\}$ and $\mathrm{Pred}_\Omega = \{a, b\}$ with $\mathrm{emb}_\Omega(b) = -\mathrm{emb}_\Omega(a)$ and $\|\mathrm{emb}_\Omega(a)\|_2 = B$, both sides equal B.

Proofs of Theorem 1, Corollary 1, and Proposition 1 are in Appendix F.

Corollary 1 (Balanced sets)**.** *If $|\,\mathrm{Pred}_\Omega\,| = |\,\mathrm{Gold}_\Omega\,|$, then $\mathrm{Recall}_\Omega = \mathrm{Precision}_\Omega$ and*

$$\|\mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) - \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega)\|_2 \le B\Big((1 - \mathrm{Recall}_\Omega) + (1 - \mathrm{Precision}_\Omega) \Big) = 2B\,(1 - \mathrm{Recall}_\Omega). \tag{22}$$

Discrete observation claims. We map this general setting to our claim-level evaluation metrics. Let Ω be the domain of atomic observation claims, with $\mathrm{emb}_{\mathrm{obs}} = \mathrm{emb}$ the text embedding, $\mathrm{Gold}_\Omega = \mathrm{Gold}^{\mathrm{obs}}$, and $\mathrm{Pred}_\Omega = \mathrm{Pred}^{\mathrm{obs}}$. Then Recall_Ω and $\mathrm{Precision}_\Omega$ are ordinary set-based observation recall and precision, and Theorem 1 gives the claim-level bound

$$\left\|\mathrm{emb}_\Omega\big(\mathrm{Gold}^{\mathrm{obs}}\big) - \mathrm{emb}_\Omega\big(\mathrm{Pred}^{\mathrm{obs}}\big)\right\|_2 \le 2B\max\big(1-\text{OBS-Recall},\ 1-\text{OBS-Precision}\big). \quad (23)$$

This bounds the distance between *averages of claim embeddings*, not between the output embeddings and ground truth embeddings. Identifying the two requires the *claim-average reading*: an output's embedding equals the average of its atomic claim embeddings. We state this as an assumption wherever it is used; text encoders satisfy it only approximately. The same theorem applies to causal claim graphs, with RC-Recall and RC-Precision on a chosen matching unit (Appendix F).

Pipeline localization. Here, we apply our error bridge theorem to the catalog coverage deficit term in Eq. (19). At the sub-system level, let $C_{\mathcal{A}(x)}$ denote the atomic claims emitted by the active factoids $\{f_i\}_{i\in\mathcal{A}(x)}$, and let $\mathrm{Recall}_{\mathcal{F}} = |C_{\mathcal{A}(x)} \cap \mathrm{Gold}(y^*)| \,/\, |\,\mathrm{Gold}(y^*)|$ be the catalog recall. Of the three stages, only the first admits a bound.

Proposition 1 (Catalog omission)**.** *Let Ω be the domain of atomic claims, and assume that* emb *is bounded by B on claims and on factoid texts. Assume further (i) the claim-average reading* $\mathrm{emb}(y^*) = \mathrm{emb}_\Omega(\mathrm{Gold}(y^*))$*, and (ii) that every matched claim $c \in C_{\mathcal{A}(x)} \cap \mathrm{Gold}(y^*)$ has* $\mathrm{emb}(c) \in$ *conv*$(D(x))$*; this holds, for instance, when each factoid emits a single atomic claim, so that its embedding is itself an element of $D(x)$. Then*

$$\|\mathrm{emb}(y^*) - \bar{v}^*\|_2 \le 2B\,(1 - \mathrm{Recall}_{\mathcal{F}}). \quad (24)$$

The factor 2 cannot be removed (Appendix F), and Appendix E computes the Stage 1 term from reference outputs alone.

Stages 2 and 3 admit no such bound. Their distances, between two points of the active convex hull and between the learned prototype and the output embedding, are not distances between set embeddings, so Theorem 1 does not apply to them. We pair them with claim metrics as a diagnostic convention: when weighting is suspected, inspect OBS-Precision; when realization is suspected, inspect RC-Precision. Table 3 in Appendix F summarizes the pairings.

7 Numerical Examples

Our formal setting predicts that a catalog of short claims enhances agent performance when claims correspond to actionable objects, subject to trade-offs in coverage, redundancy, and localization. Rather than providing a comprehensive evaluation, this section presents targeted numerical examples of our approach on a frozen model, with the slate as the only added context, to illustrate concrete areas of utility and practical limitations.

Highlights. Factoid claims are effective in software engineering when they name a file and supply the context required to use it, and in customer support when they mark a write the policy forbids. In coding tasks, a small set of descriptive claims about the target files localizes those files better than a retrieved list of bare filenames, and at least as well as giving the agent the true hindsight list of files. The advantage over the retrieved list holds on attempts that produce an action. In customer support, such claims raise correct refusals of unauthorized writes.

Conversely, claims yield diminishing returns for solution quality and for actions that change a record. In coding, a submitted patch passes about 40% of the time for every slate and both temperatures (38.8–43.5%); richer claims raise how often the agent submits, and do not raise that pass rate. In customer-support tasks that modify a booking, the claims lower success. Agents already find the target flight, and hard restrictions built from the claims block a legitimate update or fail to bind the right one.

Setup. We evaluate on a locked 50-instance slice of SWE-bench Verified (Jimenez et al., 2024; OpenAI, 2024) with an unmodified mini-SWE-agent (Yang et al., 2024), comparing an empty slate, BM25 retrieval (Robertson & Zaragoza, 2009), the catalog, and the gold file paths, and on 30 τ^2 airline tasks (Barres et al., 2026; Yao et al., 2025). Appendix G gives the datasets, metrics, and protocol.

Table 1: SWE-bench Verified, temperature 1.0, 50×10. Empty is control, BM25 is retrieval, Catalog is the factoid claims, Gold paths is the hindsight file list. Edit recall: fraction of gold source files the submitted patch touches (no patch scores 0). Excl. format fail.: that recall with format-failure runs removed (parse rate ≤ 0.05 and at least 200 steps). Resolve: fraction of runs whose patch passes the held-out tests. Submit: fraction of runs that turn in a patch. Resolve | submit: pass rate among those patches. Edit recall, resolve, and submission are means over ten repeats; Resolve | submit is pooled over 500 runs, so it multiplies with submission to give resolve. **Bold** highest, *italic* second (a tie is marked on both). * Holm $p < 0.05$ versus empty. † $p < 0.05$ versus BM25. ‡ Wilcoxon $p < 0.05$ versus empty, bootstrap interval includes 0. Resolve | submit is unmarked.

Arm	Edit recall	Excl. format fail.	Resolve	Submit	Resolve|submit
Empty slate	0.497	0.578	0.260	0.652	0.399
BM25 k=5	0.559	0.601	*0.310*‡	*0.712*	0.435
Catalog	**0.616***	**0.673***†	**0.320**‡	**0.740**	0.432
Gold paths	*0.598*	*0.654*	0.284	*0.712*	0.399

Table 2: τ^2 airline, one run per arm. Primary is refusal or write-entity recall. **Bold** highest, *italic* second, a tie marked on both. One run, so no significance mark.

Split	Empty	Catalog	Gold entities
Refuse, n=10, primary	0.70	**1.00**	*0.80*
Refuse, official	$7/10$	**10/10**	*8/10*
Write, n=20, primary	*0.93*	0.88	**0.94**
Write, official	**11/20**	$9/20$	**11/20**

Results. On all runs, catalog versus empty is $+0.118$ ($p = 0.00037$, Holm 0.00073; CI $[+0.028, +0.148]$).

Analysis. *Retrieval versus catalog.* Empty $<$ BM25 $<$ catalog holds in 6 of 10 repeats. On all runs, BM25 minus empty is $+0.062$ ($p = 0.052$) and catalog minus BM25 is $+0.057$ ($p = 0.054$). After stuck runs are removed, those gaps are $+0.029$ ($p = 0.30$) and $+0.063$ ($p = 0.046$). BM25's edge is mostly fewer stuck runs. The catalog's edge over BM25 remains on recall and is $+0.010$ on resolve ($p = 1$).

Appendix G analyzes format failures, tests the split between submitting and resolving with a forecast made before the temperature-0 scores existed, and reports the airline ablations.

8 Concluding Remarks

In this work, we introduced *sparse context synthesis*, a formal framework for agentic harness optimization where a catalog of gated factoids generates targeted claims for an episode, which an off-the-shelf agent then synthesizes into a final output. Our framework evaluates a catalog based on the mutual information its claims share with an episode's latent target, certified from below by the Barber–Agakov bound. This approach allows us to localize omitted information and formally decouple the design pressures of coverage, redundancy, and localization; notably, we show that negative redundancy exposes structural synergy that naive pruning would destroy. Furthermore, we established a theoretical error bridge that connects claim-level recall and precision to embedding-space error with a tight constant and bounds the error caused by catalog omissions.

Empirically, our evaluations on SWE-bench Verified and τ^2-bench highlight a critical operational dichotomy: while injected claims successfully guide the agent to localize relevant files and adhere to policy constraints (e.g., refusing forbidden writes), metrics that depend heavily on the agent's generative realization, such as final patch correctness and record updates, do not uniformly improve. Currently, our analysis treats evaluation as set retrieval, relies on a claim-average reading for the omission bound, and provides primarily diagnostic insights for the weighting and realization stages. Formalizing bounds for these latter stages and testing our design principles on larger, more complex catalogs remain compelling directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for writing, code generation, contextual background research, and validation of theoretical claims.

We have not used generative AI tools for the core research questions, the core ideas, motivation, theoretical claims, choice of data sets, design of numerical experiments.

We have reviewed and thoroughly verified all AI-assisted work, such as code, text generation (writing), contextual background research, and validation of theoretical claims. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Standing definitions are in Section 2. Catalog-fit diagnostics are in Section 5, with estimators in Appendix E. The error bridge is Theorem 1; the equal-size case is Corollary 1; the catalog-omission bound is Proposition 1; Appendix F restates all three with complete proofs. The factoid catalogs used in Section 7 are specified in Appendix H, and the protocol in Appendix G.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2000.

David Barber and Felix V. Agakov. The IM algorithm: A variational approach to information maximization. In *Advances in Neural Information Processing Systems*, 2003.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. In *International Conference on Machine Learning*, 2026.

Ron Begleiter, Katya Egert-Berg, Gilad Saban, and Gil Shabat. Loom: Weaving diagnostic strands into free-text consensus via embedding-space reweighting. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026. URL `https://arxiv.org/abs/2609.02649`. To appear. arXiv:2609.02649.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapters 9–10.

Qianshu Cai, Yonggang Zhang, Jun Nie, Maohao Ran, Huajiang Zheng, Jun Song, Xinmei Tian, Yike Guo, and Wei Xue. Safe harness self-evolution: A theoretical analysis of feasibility and limits. *arXiv preprint arXiv:2609.08175*, 2026.

Ashwin Gerard Colaco and Nada Lahjouji. What to keep, what to forget: A rate–distortion view of memory compaction in LLMs and agents. *arXiv preprint arXiv:2607.08032*, 2026.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pp. 334–343, 1997.

Hai Huang. Directed information γ-covering: An information-theoretic framework for context engineering. *arXiv preprint arXiv:2510.00079*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

Mintong Kang, Nezihe Merve Gürel, Ning Yu, Dawn Song, and Bo Li. C-RAG: Certified generation risks for retrieval-augmented language models. In *International Conference on Machine Learning*, 2024.

Hyunjun Kim. Entropic context shaping: Information-theoretic filtering for context-aware LLM agents. *arXiv preprint arXiv:2601.11585*, 2026.

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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*, 2025.

Qianyu Meng, Yanan Wang, Liyi Chen, Yihang Li, Wei Wu, Wenyuan Jiang, Qimeng Wang, Chengqiang Lu, Yan Gao, Yi Wu, and Yao Hu. Agent harness for large language model agents: A survey. *Preprints*, 2026. doi: 10.20944/preprints202604.0428.v3. URL `https://www.preprints.org/manuscript/202604.0428/v3`.

OpenAI. Introducing SWE-bench Verified. `https://openai.com/index/introducing-swe-bench-verified/`, 2024.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, 2016.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 2009. doi: 10.1561/1500000019.

Chengshuai Shi, Kun Yang, Zihan Chen, Jundong Li, Jing Yang, and Cong Shen. Efficient prompt optimization through the lens of best arm identification. In *Advances in Neural Information Processing Systems*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106, 2026.

Hayder Tirmazi, Sam Markelon, Allison Bishop, and Michael Mitzenmacher. Context compaction theory. *arXiv preprint arXiv:2608.01326*, 2026.

Paroma Varma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. Learning dependency structures for weak supervision models. In *International Conference on Machine Learning*, 2019.

C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1): 95–103, 1983.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, pp. 74824–74876, 2025.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eC4ygDs02R.

A Related Work

Context adaptation. An off-the-shelf agent can improve by changing its context rather than its weights: by reflecting on failed attempts (Shinn et al., 2023), keeping a memory that grows at test time (Suzgun et al., 2026), or evolving its prompt from execution traces (Agrawal et al., 2026). Agentic Context Engineering (ACE; Zhang et al., 2026) keeps the context as a playbook of strategies and edits it by small incremental updates rather than full rewrites. Two failure modes observed in experiments motivate this design: rewriting the whole context erases accumulated detail (context collapse), and optimizers that favor short prompts drop domain-specific tactics (brevity bias).

Information-theoretic view. For claims computed from the current episode, both failure modes have a formal counterpart. By the chain rule and the data-processing inequality (Cover & Thomas, 2006), adding a claim cannot decrease the mutual information the context carries about the episode's latent target, and rewriting the context as a function of itself cannot increase it. Section 5 measures catalog fit by this mutual information. A playbook shared across episodes carries no information about a new episode's target; it helps by changing how the agent uses the evidence. Mutual information does not account for whether a bounded agent can extract what the context carries. Usable information does, and unlike mutual information it can be increased by computation (Xu et al., 2020), so a rewrite that adds no information can still help. This is why Section 6 separates the catalog's omissions, which we bound, from the agent's realization, which we diagnose.

Theory of harness optimization. Existing theory analyzes the search over harnesses: prompt selection under an evaluation budget as best-arm identification (Shi et al., 2024), and when a self-modification of an agent with a frozen model can be certified and safely adopted (Cai et al., 2026). For the context itself, Mei et al. (2025) pose context engineering as optimizing the functions that generate the context and frame retrieval as maximizing mutual information with the target answer. Other work selects or compresses a given pool, by covering it with directed information (Huang, 2025), by relating compaction to one-way communication complexity (Tirmazi et al., 2026), or by treating memory compaction as rate–distortion, with a Fano lower bound on task error (Colaco & Lahjouji, 2026). Entropic context shaping scores a passage by how far it moves the model's answer distribution toward the correct answer (Kim, 2026), a model-side counterpart of the per-episode shift we measure under the true distribution (Section 5). Conformal analysis certifies the risk of retrieval-augmented generation (Kang et al., 2024). These works do not analyze context that a catalog of gated functions computes for each episode, which is our setting.

Retrieval and long contexts. Retrieval-augmented generation conditions the model on passages retrieved by similarity to a query (Lewis et al., 2020). A factoid instead computes a short claim from the episode, such as where the identifiers named in an issue are defined, and fires only when its gate applies. Our SWE-bench Verified experiments include a BM25 retrieval arm for comparison (Section 7). More context is not always better: models are distracted by irrelevant context (Shi et al., 2023) and use information in the middle of long inputs less reliably (Liu et al., 2024), which is one reason to keep activation sparse.

Weak supervision. Factoids resemble the labeling functions of weak supervision: noisy, overlapping heuristics that abstain when they do not apply. Label models combine such sources without ground truth by fitting a factorized generative model of their outputs given a latent label, from EM estimates of annotator error rates (Dawid & Skene, 1979) to data programming (Ratner et al., 2016). Correlated sources double-count their evidence unless their dependencies are modeled, and the dependencies can be learned from unlabeled data (Varma et al., 2019). Consensus weighting (Begleiter et al., 2026) applies this idea to free-text claims in embedding space. Our redundancy measure (Section 5.2) quantifies the overlap as information about the target; it is negative when factoids are informative only jointly.

Online learning. The consensus weights are normalized over the factoids that fire, as in prediction with specialists, where experts may abstain on a round and the weights are renormalized over those that predict (Freund et al., 1997). Specialist algorithms learn the weights from losses observed after each round; consensus weighting sets them within an episode from agreement among the claims, without feedback.

B Variational Stand-In and Entropy Conventions

The table contrasts the generative parameters θ with the variational stand-in q of Section 2.

	θ	q
What it is	parameters of $p_\theta(X, Z)$	a stand-in for $p_\theta(Z \mid X)$
Lives on	how targets generate observations	beliefs about Z after seeing X (or factoids)
Updated by	M-step	E-step: $q \leftarrow p_{\theta^{(t)}}(Z \mid X)$ when tractable
Tight when	the model matches the data	$q(Z \mid X) = p_\theta(Z \mid X)$

B.1 Entropy, mutual information, and what zero means

Unconditional entropy $H(Z)$ is residual unpredictability of the target before any observation; under the model it is $H_\theta(Z)$, the entropy of the prior $p_\theta(Z)$. Conditional entropy under the model is the average posterior uncertainty after seeing X,

$$H_\theta(Z \mid X) = \mathbb{E}_{x \sim p_\theta(X)}\big[H\big(p_\theta(Z \mid x)\big)\big] = -\mathbb{E}_{p_\theta(x,z)}\big[\log p_\theta(z \mid x)\big]. \tag{25}$$

Mutual information under the model is the reduction in that uncertainty:

$$I_\theta(X; Z) = H_\theta(Z) - H_\theta(Z \mid X). \tag{26}$$

Thus $H_\theta(Z \mid X)$ is the part of $H_\theta(Z)$ that the observation *fails* to remove. The Barber–Agakov bound (Section 4) lower-bounds I_θ by substituting q for the intractable posterior; the bound is tight iff $q(Z \mid X) = p_\theta(Z \mid X)$.

If $H_\theta(Z \mid X) = 0$, then X determines Z: for almost every x the posterior is a point mass, equivalently $Z = g(X)$ almost surely, and $I_\theta(X; Z) = H_\theta(Z)$. In words, the full observation identifies the target with no leftover ambiguity. This is *not* the same as "Z and X share the same information":

- $H(Z \mid X) = 0$ means X determines Z (the observation is sufficient for the target).
- $H(X \mid Z) = 0$ means Z determines X (the target reconstructs the whole observation). Almost never true.
- The two variables carry the same information only if both vanish, i.e. $I(X; Z) = H(X) = H(Z)$ (a bijection up to renaming).

Zero conditional entropy is asymmetric: a long log or a large repository can pin down a single target while remaining a high-entropy object given that target. In the catalog setting the operational residual is usually $H(Z \mid S_{\mathcal{F}}(X))$, not $H(Z \mid X)$. Even if the full observation determines Z, factoids can leave residual entropy; that leftover is the coverage gap $\Gamma(\mathcal{F})$ of Section 5.

C Generative Setup and EM

C.1 Probability Space and Generative Setup

We model the observed episode as a random variable X taking values in $\mathcal{X}$, and the unobserved target as a latent random variable Z taking values in $\mathcal{Z} = \{1, \ldots, K\}$, the set of K discrete latent states; both are defined on a common probability space.

Note on the size of K. In realistic tasks the target space can be very large: the files of a repository, the records of a customer database, or a fine-grained taxonomy of failure causes all expand the global label space, and it may keep growing as the environment evolves. Importantly, K is a modeling choice rather than a fixed constant: coarse buckets keep K modest, while fine-grained labels drive K into the large-cardinality regime. In practice one rarely runs inference against the full set $\{1, \ldots, K\}$ for a single episode. The active index set $\mathcal{A}(x)$ already restricts attention to context-relevant channels, so the posterior typically concentrates on an episode-local support of size $K_x \ll K$; hierarchical taxonomies and candidate generation further keep EM tractable despite a large global K.

For an observed condition $X = x$, the active language functionals $\{f_i\}_{i \in \mathcal{A}(x)}$ generate an emission vector of text outputs whose continuous embeddings form $D(x) = \{v_i(x)\}_{i \in \mathcal{A}(x)} \subset \mathbb{R}^d$. The joint distribution governed by parameters $\theta \in \Theta$ factorizes as $p_\theta(X, Z) = p_\theta(Z)\, p_\theta(X \mid Z)$.

C.2 Variational Lower Bound (ELBO) and Convergence

The observed log-likelihood $L(\theta) = \log p_\theta(X) = \log \sum_{z \in \mathcal{Z}} p_\theta(X, z)$ is intractable to optimize directly due to the summation inside the logarithm. For any variational distribution $q(Z \mid X)$ over latent targets, Jensen's inequality on the concave logarithm yields the Evidence Lower Bound (ELBO), denoted $\mathcal{L}(q, \theta)$ (Bishop, 2006):

$$\begin{aligned} \log p_\theta(X) &= \log \sum_{z \in \mathcal{Z}} q(z \mid X) \frac{p_\theta(X, z)}{q(z \mid X)} \\ &\geq \sum_{z \in \mathcal{Z}} q(z \mid X) \log \frac{p_\theta(X, z)}{q(z \mid X)} \equiv \mathcal{L}(q, \theta). \end{aligned} \tag{27}$$

Expanding $\mathcal{L}(q, \theta)$ yields the fundamental identity linking marginal likelihood, the ELBO, and Kullback-Leibler (KL) divergence:

$$\log p_\theta(X) = \mathcal{L}(q, \theta) + D_{\mathrm{KL}}\Big(q(Z \mid X) \;\|\; p_\theta(Z \mid X)\Big). \tag{28}$$

Since $D_{\mathrm{KL}}(\cdot \,\|\, \cdot) \geq 0$, the iterative EM algorithm (Dempster et al., 1977; Bishop, 2006) guarantees monotonic improvement of the observed likelihood via coordinate ascent on $\mathcal{L}(q, \theta)$:

- **E-Step (Posterior Computation):** Fix $\theta^{(t)}$ and update q:

$$\begin{aligned} q^{(t+1)}(Z \mid X) &= p_{\theta^{(t)}}(Z \mid X) \\ &\implies D_{\mathrm{KL}}\big(q^{(t+1)} \,\|\, p_{\theta^{(t)}}\big) = 0 \\ &\implies \mathcal{L}(q^{(t+1)}, \theta^{(t)}) = \log p_{\theta^{(t)}}(X). \end{aligned} \tag{29}$$

- **M-Step (Parameter Maximization):** Fix $q^{(t+1)}$ and maximize $\mathcal{L}$ with respect to θ:

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} \sum_{z \in \mathcal{Z}} q^{(t+1)}(z \mid X) \log p_\theta(X, z). \tag{30}$$

This ensures non-decreasing likelihood at every iteration:

$$\log p_{\theta^{(t+1)}}(X) \geq \mathcal{L}(q^{(t+1)}, \theta^{(t+1)}) \geq \mathcal{L}(q^{(t+1)}, \theta^{(t)}) = \log p_{\theta^{(t)}}(X). \tag{31}$$

Under standard regularity conditions the iterates converge to a stationary point of $L(\theta)$ (Wu, 1983); global optimality is not guaranteed.

Reading the Barber–Agakov bound. This continues the reading of (10) in Section 4. A predictor that is always certain and right has surprise 0 and certifies all of $H_\theta(Z)$; one that ignores the observation and predicts the prior has surprise $H_\theta(Z)$ and certifies nothing. Since no predictor is less surprised on average than the true posterior, the certified reduction can only understate what the observation reveals; the shortfall is exactly $\mathbb{E}_x\big[D_{\mathrm{KL}}\big(p_\theta(Z \mid x) \,\|\, q(Z \mid x)\big)\big]$. The EM E-step sets $q \leftarrow p_{\theta^{(t)}}(Z \mid X)$ and therefore saturates (10) at the current parameters. The M-step then increases the marginal likelihood, which need not increase $I_\theta(X; Z)$: likelihood and mutual information are different objectives, and $H_\theta(Z)$ itself moves with θ. What the E-step does guarantee is that, at the current parameters, the posterior extracts all the information about Z that the model assigns to X. The variational distribution $\beta(J \mid X)$ in that procedure plays an analogous role when the latent of interest is which channel carries the consensus signal: factoids whose embeddings align with c^* receive higher mass, while outliers are down-weighted.

D Information Geometry of Consensus Weighting

Information geometry (m-projection centroid). Each functional channel $f_i(x)$ induces a candidate posterior $p_i(Z \mid X)$ on $\mathcal{Z}$ (e.g., by restricting the likelihood to that channel's evidence). Weighting the channels by $w(x)$, the consensus model minimizes their total forward KL discrepancy:

$$\begin{aligned} \theta^* &\in \arg\min_{\theta \in \Theta} \sum_{i \in \mathcal{A}(x)} w_i(x)\, D_{\mathrm{KL}}\big(p_i(Z \mid X) \,\big\|\, p_\theta(Z \mid X)\big) \\ &= \arg\min_{\theta \in \Theta} D_{\mathrm{KL}}\big(\bar{p}_w(Z \mid X) \,\big\|\, p_\theta(Z \mid X)\big), \end{aligned} \tag{32}$$

where $\bar{p}_w := \sum_{i \in \mathcal{A}(x)} w_i(x)\, p_i$ is the w-weighted mixture of the channel posteriors. The two objectives differ by $\sum_i w_i(x) \sum_z p_i(z) \log p_i(z) - \sum_z \bar{p}_w(z) \log \bar{p}_w(z)$, the w-weighted Jensen–Shannon divergence among the channel posteriors, which does not depend on θ. Hence the consensus posterior $p_{\theta^*}(Z \mid X)$ is the m-projection (mixture projection) of $\bar{p}_w$ onto the model family $\{p_\theta\}$ (Amari & Nagaoka, 2000). The weights matter only through this mixture; with uniform weights it is the plain average of the channel posteriors. The mixture $\bar{p}_w$ combines channel posteriors with the same weights with which the semantic centroid (4) combines channel embeddings, and the consensus centroid c^* in (6) is the corresponding construction in embedding space. The correspondence is an analogy, not a derivation: neither construction determines the other.

D.1 Statistical-geometry counterpart

Each active factoid induces a candidate posterior $p_i(Z \mid X)$ (or an embedding $v_i(x)$ as a continuous proxy). Catalog geometry is the set of such channels on the statistical manifold of distributions on $\mathcal{Z}$ (Amari & Nagaoka, 2000). The m-projection centroid (32) and the factoid centroid (6) aggregate these channels; fit means that, for typical $x \sim P_X$, the channel set spans the posterior mass of Z tightly enough that the consensus prototype is close (in KL or embedding distance) to the oracle posterior/prototype. Thus information utility $U(\mathcal{F})$ is the objective, and statistical geometry describes the space in which consensus is computed.

E Empirical Estimators for Catalog Fit

On a finite sample $\mathcal{X}_N = \{x_j\}_{j=1}^N$, materialize (14)–(16) as follows.

Labeled or oracle setting. When target labels z_j (or gold claims from an evaluation oracle) are available, estimate discrete posteriors by frequencies or a calibrated classifier $\hat{q}(Z \mid S_{\mathcal{F}}(x))$, and set

$$\hat{U}(\mathcal{F}) = \hat{H}(Z) - \frac{1}{N}\sum_{j=1}^{N} \hat{H}\big(Z \mid S_{\mathcal{F}}(x_j)\big), \qquad \hat{\Gamma}(\mathcal{F}) = \hat{I}(Z;X) - \hat{U}(\mathcal{F}). \tag{33}$$

When gold claims are available, the catalog recall deficit $1 - \mathrm{Recall}_{\mathcal{F}}$ of Section 6 is a practical, measurable companion to $\hat{\Gamma}$. It is a heuristic proxy for coverage, not an estimator of Γ: no formal relation between the two is shown here, although Proposition 1 ties the recall deficit to the catalog-omission term in embedding space.

Variational / label-lean setting. Absent full labels, maximize a Barber-Agakov or ELBO-style lower bound on $I(Z; S_{\mathcal{F}}(X))$ (Barber & Agakov, 2003; Bishop, 2006) with variational $q(Z \mid S_{\mathcal{F}}(X))$, exactly as in Section 4. The distribution $\beta(J \mid X)$ supplies a cheap surrogate responsibility layer for channel selection inside that bound.

Embedding-space residual. Averaged over episodes, the Stage 1 term is directly computable whenever reference outputs exist. By (17), its average over a sample $\mathcal{X}_N = \{x_j\}_{j=1}^{N}$ is

$$\overline{\delta} = \frac{1}{N}\sum_{j=1}^{N} \min_{w \in \Delta^{|\mathcal{A}(x_j)|-1}} \Big\| \mathrm{emb}(y^*(x_j)) - \sum_{i \in \mathcal{A}(x_j)} w_i \, v_i(x_j) \Big\|_2, \tag{34}$$

where each term is a small convex program over the simplex. Unlike (24), it needs reference outputs but no claim labels; a large $\overline{\delta}$ flags episodes whose active factoids cannot express the reference.

When claim labels are scarce, the embedding-space residual $\overline{\delta}$ of (34) is a practical proxy for catalog omission.

F Proofs and Causal Claim Graphs

This appendix restates and proves the results of Section 6 and applies the error bridge to causal claim graphs.

Generic setting. Let Ω be a domain of embeddable evaluative objects, and let $\mathrm{emb}_\Omega : \Omega \to \mathbb{R}^d$ map each object into a semantic vector space with uniformly bounded norms $\|\mathrm{emb}_\Omega(\omega)\|_2 \le B$. Given a gold collection $\mathrm{Gold}_\Omega \subseteq \Omega$ and a predicted collection $\mathrm{Pred}_\Omega \subseteq \Omega$, define the set embeddings as compositional averages

$$\begin{aligned} \mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) &= \frac{1}{|\,\mathrm{Gold}_\Omega\,|} \sum_{\omega \in \mathrm{Gold}_\Omega} \mathrm{emb}_\Omega(\omega), \\ \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega) &= \frac{1}{|\,\mathrm{Pred}_\Omega\,|} \sum_{\omega' \in \mathrm{Pred}_\Omega} \mathrm{emb}_\Omega(\omega'). \end{aligned} \tag{20}$$

and partition into true, missed, and false objects:

$$\Omega_{\mathrm{true}} = \mathrm{Pred}_\Omega \cap \mathrm{Gold}_\Omega, \quad \Omega_{\mathrm{missed}} = \mathrm{Gold}_\Omega \setminus \mathrm{Pred}_\Omega, \quad \Omega_{\mathrm{false}} = \mathrm{Pred}_\Omega \setminus \mathrm{Gold}_\Omega\,.$$

The vector difference then admits the same partition identity used for claim sets:

$$\begin{aligned} \mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) - \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega) = \Big(\tfrac{1}{|\,\mathrm{Gold}_\Omega\,|} - \tfrac{1}{|\,\mathrm{Pred}_\Omega\,|}\Big) \sum_{\omega \in \Omega_{\mathrm{true}}} \mathrm{emb}_\Omega(\omega) \\ + \tfrac{1}{|\,\mathrm{Gold}_\Omega\,|} \sum_{\omega \in \Omega_{\mathrm{missed}}} \mathrm{emb}_\Omega(\omega) - \tfrac{1}{|\,\mathrm{Pred}_\Omega\,|} \sum_{\omega \in \Omega_{\mathrm{false}}} \mathrm{emb}_\Omega(\omega). \end{aligned} \tag{35}$$

Theorem 1 (Error bridge)**.** *Let Ω be any domain of embeddable objects as above, and let* Gold_Ω *and* Pred_Ω *be finite nonempty subsets of Ω, with*

$$\mathrm{Recall}_\Omega = \frac{|\Omega_{\mathrm{true}}|}{|\,\mathrm{Gold}_\Omega\,|}, \qquad \mathrm{Precision}_\Omega = \frac{|\Omega_{\mathrm{true}}|}{|\,\mathrm{Pred}_\Omega\,|}.$$

Then

$$\begin{aligned}\|\mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) - \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega)\|_2 \le B\Big(&\big|\mathrm{Recall}_\Omega - \mathrm{Precision}_\Omega\big| \\ &+ (1-\mathrm{Recall}_\Omega) + (1-\mathrm{Precision}_\Omega)\Big) \\ &= 2B\max\big(1-\mathrm{Recall}_\Omega,\ 1-\mathrm{Precision}_\Omega\big).\end{aligned} \tag{21}$$

The bound is attained: for $\mathrm{Gold}_\Omega = \{a\}$ *and* $\mathrm{Pred}_\Omega = \{a, b\}$ *with* $\mathrm{emb}_\Omega(b) = -\mathrm{emb}_\Omega(a)$ *and* $\|\mathrm{emb}_\Omega(a)\|_2 = B$*, both sides equal* B*.*

Proof of Theorem 1. Let $t = |\Omega_{\mathrm{true}}|$, so that $|\Omega_{\mathrm{missed}}| = |\,\mathrm{Gold}_\Omega\,| - t$ and $|\Omega_{\mathrm{false}}| = |\,\mathrm{Pred}_\Omega\,| - t$. Apply the triangle inequality to the three sums in (35) and bound each embedding norm by B. The matched sum contributes $t\,\big|\frac{1}{|\,\mathrm{Gold}_\Omega\,|} - \frac{1}{|\,\mathrm{Pred}_\Omega\,|}\big|\,B = |\mathrm{Recall}_\Omega - \mathrm{Precision}_\Omega|\,B$, the missed sum contributes $\frac{|\Omega_{\mathrm{missed}}|}{|\,\mathrm{Gold}_\Omega\,|}\,B = (1-\mathrm{Recall}_\Omega)\,B$, and the false sum contributes $\frac{|\Omega_{\mathrm{false}}|}{|\,\mathrm{Pred}_\Omega\,|}\,B = (1-\mathrm{Precision}_\Omega)\,B$. Summing gives the first line of (21). The second follows from $|r-p| + (1-r) + (1-p) = 2 - 2\min(r,p) = 2\max(1-r,\,1-p)$. □

Corollary 1 (Balanced sets)**.** *If* $|\,\mathrm{Pred}_\Omega\,| = |\,\mathrm{Gold}_\Omega\,|$*, then* $\mathrm{Recall}_\Omega = \mathrm{Precision}_\Omega$ *and*

$$\|\mathrm{emb}_\Omega(\mathrm{Gold}_\Omega) - \mathrm{emb}_\Omega(\mathrm{Pred}_\Omega)\|_2 \le B\Big((1-\mathrm{Recall}_\Omega) + (1-\mathrm{Precision}_\Omega)\Big) = 2B\,(1-\mathrm{Recall}_\Omega). \tag{22}$$

Proof of Corollary 1. With $n = |\,\mathrm{Gold}_\Omega\,| = |\,\mathrm{Pred}_\Omega\,|$, both recall and precision equal t/n, so the first term of (21) vanishes. □

When the two sets have the same size, both averages divide by the same number, so the claims they share enter with equal weight and cancel exactly. What remains is the missed claims pulling the gold average one way and the spurious claims pulling the predicted average another; since there are exactly as many misses as spurious claims, the two deficits coincide. When the sizes differ, the shared claims carry weight $1/|\,\mathrm{Gold}_\Omega\,|$ in one average and $1/|\,\mathrm{Pred}_\Omega\,|$ in the other and no longer cancel; the term $|\mathrm{Recall}_\Omega - \mathrm{Precision}_\Omega|$ in (21) pays for this imbalance. In particular, (22) does not extend to approximately balanced sets: in the example of Theorem 1 the sizes differ by one, and the distance B is twice the balanced bound $B/2$.

Proposition 1 uses the pipeline quantities of Section 6: $C_{\mathcal{A}(x)}$ denotes the atomic claims emitted by the active factoids $\{f_i\}_{i\in\mathcal{A}(x)}$, $\mathrm{Recall}_{\mathcal{F}} = |C_{\mathcal{A}(x)} \cap \mathrm{Gold}(y^*)|\,/\,|\,\mathrm{Gold}(y^*)|$ is the catalog recall, and $\bar{v}^*$ is the oracle centroid (17), the point of $\mathrm{conv}(D(x))$ closest to $\mathrm{emb}(y^*)$.

Proposition 1 (Catalog omission)**.** *Let* Ω *be the domain of atomic claims, and assume that* emb *is bounded by* B *on claims and on factoid texts. Assume further (i) the claim-average reading* $\mathrm{emb}(y^*) = \mathrm{emb}_\Omega(\mathrm{Gold}(y^*))$*, and (ii) that every matched claim* $c \in C_{\mathcal{A}(x)} \cap \mathrm{Gold}(y^*)$ *has* $\mathrm{emb}(c) \in$ *conv*$(D(x))$*; this holds, for instance, when each factoid emits a single atomic claim, so that its embedding is itself an element of* $D(x)$*. Then*

$$\|\mathrm{emb}(y^*) - \bar{v}^*\|_2 \le 2B\,(1-\mathrm{Recall}_{\mathcal{F}}). \tag{24}$$

Proof of Proposition 1. If no gold claim is matched, then $\mathrm{Recall}_{\mathcal{F}} = 0$, and the bound $2B$ holds because both points have norm at most B. Otherwise let $T = C_{\mathcal{A}(x)} \cap \mathrm{Gold}(y^*)$ be the matched claims. By (ii) their average $\mathrm{emb}_\Omega(T)$ lies in $\mathrm{conv}(D(x))$, so by (17) and (i), $\|\mathrm{emb}(y^*) - \bar{v}^*\|_2 \le \|\mathrm{emb}_\Omega(\mathrm{Gold}(y^*)) - \mathrm{emb}_\Omega(T)\|_2$. Apply Theorem 1 with $\mathrm{Gold}_\Omega = \mathrm{Gold}(y^*)$ and $\mathrm{Pred}_\Omega = T$: since $T \subseteq \mathrm{Gold}(y^*)$, $\mathrm{Precision}_\Omega = 1$ and $\mathrm{Recall}_\Omega = \mathrm{Recall}_{\mathcal{F}}$, and (21) gives (24). □

The factor 2 cannot be removed: with $\mathrm{Gold}(y^*) = \{a, b\}$, a single matched claim a, $D(x) = \{\mathrm{emb}(a)\}$, $\mathrm{emb}(b) = -\mathrm{emb}(a)$ and $\|\mathrm{emb}(a)\|_2 = B$, the left side equals B while $\mathrm{Recall}_{\mathcal{F}} = 1/2$, so (24) holds with equality.

Causal claim graphs. Causal conclusions are not a flat bag of claims: they are a graphical structure over claims. Let $G^* = (V^*, R^*)$ and $\hat{G} = (\hat{V}, \hat{R})$ denote the gold and predicted causal claim graphs. Fix a matching unit $\Omega = \mathcal{E}$ on this graph layer, e.g. directed causal edges, attributed support relations, or causal paths, and let $\mathrm{emb}_{\mathrm{rc}} : \mathcal{E} \to \mathbb{R}^{d'}$, with $\|\mathrm{emb}_{\mathrm{rc}}(\cdot)\|_2 \le B_{\mathrm{rc}}$, embed each graphical object into a semantic geometry of causal structure. Applying Theorem 1 at this layer yields

$$\|\mathrm{emb}_{\mathrm{rc}}(R^*) - \mathrm{emb}_{\mathrm{rc}}(\hat{R})\|_2 \le 2B_{\mathrm{rc}} \max\big(1 - \text{RC-Recall},\ 1 - \text{RC-Precision}\big), \tag{36}$$

where RC-Recall and RC-Precision are the natural graphical analogues of set recall/precision on the chosen matching unit. Thus observation precision and causal precision are not ad hoc proxies: they are parallel specializations of the same error-bridge machinery, differing only in the domain Ω being embedded.

Pipeline stages and claim metrics. Table 3 pairs each stage of the error decomposition in Section 6 with the claim metric used to diagnose it.

Table 3: Pipeline stages, their vector-space terms, and the claim metrics paired with them. Only Stage 1 and the end-to-end row carry a bound; the Stage 2 and Stage 3 pairings are diagnostic conventions. In the end-to-end row, Ω is the domain of atomic claims.

Stage	Vector metric (L_2)	Claim metric	Relation
1: Omission	$\|\mathrm{emb}(y^*) - \bar{v}^*\|_2$	$1 - \mathrm{Recall}_{\mathcal{F}}$	Bound $\le 2B(1 - \mathrm{Recall}_{\mathcal{F}})$, Prop. 1
2: Weighting	$\|\bar{v}^* - \bar{v}_{\text{learned}}\|_2$	$1 -$ OBS-Precision	Diagnostic convention (no bound)
3: Realization	$\|\bar{v}_{\text{learned}} - \mathrm{emb}(\hat{y})\|_2$	$1 -$ RC-Precision	Diagnostic convention (no bound)
End-to-end	$\|\mathrm{emb}(y^*) - \mathrm{emb}(\hat{y})\|_2$	$\max(1 - \mathrm{Recall}_\Omega,\ 1 - \mathrm{Precision}_\Omega)$	Bound $\le 2B \max(\cdot)$, Thm. 1; claim-average reading

G Experimental Details

Datasets. SWE-bench Verified (Jimenez et al., 2024; OpenAI, 2024): 50 locked instances (test split, sorted by instance id, seed 42, first 50; django is 31). Unmodified mini-SWE-agent (Yang et al., 2024). Primary metric is agent-edit file recall: the fraction of gold source files that the submitted patch also touches. No patch scores zero. Secondary: slate recall, and resolve (whether the patch passes the held-out tests). Slate recall is fixed by the builder: empty 0, BM25 0.530, catalog 0.685, gold paths 1.

τ^2 airline (Barres et al., 2026; Yao et al., 2025): 10 refuse tasks and 20 write tasks, the unmodified τ^2 `LLMAgent`, one run per arm. Primary: refusal correctness, or write-entity recall. Secondary: $\mathrm{pass}^1 = \mathrm{DB} \times \mathrm{COMMUNICATE}$.

Experimental protocol. Four SWE slates. Empty omits the block. BM25 (Robertson & Zaragoza, 2009) injects five paths (Okapi, k=5; the query is the issue text). The catalog injects gated, ranked factoids (median 1,206 characters, five factoids). Gold paths inject every gold file path and no hunks (median 254 characters). Shared settings: the slice, the catalog, the scaffold, 250 steps, a $3 cap, and `nvidia/nemotron-3.5-lightning`. One study is temperature 0; the other is temperature 1.0, top-p 0.95. Each is ten repeats (2,000 runs) plus 2,000 resolve evaluations (`swebench` 3.0.17; six harness errors at temperature 1.0, four at temperature 0). The temperature-0 scores were computed after Table 4 was hashed.

The test unit is the instance mean over ten repeats: paired Wilcoxon, Holm on the two catalog-versus-empty recall tests, and a cluster bootstrap over ten repository families. Excluding format failures, catalog minus empty is $+0.094$ on the repeat-level macros and $+0.092$ on the instance means used in the test. A format-failure run has parse rate ≤ 0.05 and at least 200 steps. Mean parse rate is quoted beside it, because the flag misses runs that waste turns and then recover. Empty and catalog ran together; BM25 and gold paths ran together later the same day. The cross-pair comparisons span that gap. Endpoint latency and error rate were flat. The airline oracle injects gold entity ids and wiki text, and withholds write payloads. A gain that drops a control pass fails gold-keep.

Table 4: Temperature-0 resolve on the same slice and arms as Table 1, forecast before the scores existed. Predicted is temperature-0 submission times the temperature-1.0 pass rate of a submitted patch. Tolerance ± 0.03.

Arm	Predicted resolve	Actual resolve	Actual − predicted
Empty slate	0.243	0.250	+0.007
BM25	0.295	0.278	−0.017
Catalog	0.315	0.302	−0.013
Gold paths	0.283	0.274	−0.009

Format failures. Format-failure rates at temperature 1.0 are 0.138 (empty), 0.070 (BM25), 0.084 (catalog), 0.084 (gold paths); at temperature 0, empty was 0.196 and the catalog 0.096. Empty-slate mean parse rate at temperature 1.0 is 0.795. The empty rate is 1.6 to 2 times any populated slate, and which populated slate is lowest swaps with temperature. Of the empty slate's stuck runs, 65% open by echoing the `<format_example>` wrapper. The endpoint returns `reasoning_content` apart from `content`; the scaffold reads only `content`, which comes back empty when the token budget is spent on reasoning. Temperature cuts the empty slate's stuck-run rate by about a third. The +0.092 instance-level gap is what remains after those runs are removed.

Resolve versus submit. We investigate the relationship between the agent's success in submitting a patch and its success in resolving the task. In Table 1, resolve is submission times the pass rate of a submitted patch. On those same runs the product always matches, by arithmetic, so temperature 1.0 alone cannot test the split. What it shows is that the pass rate sits near 40% for every slate, from the empty slate to the true file list, while the submission rate changes. A better slate helps the agent submit a patch. It does not make a submitted patch more likely to pass. Temperature 0 is a second draw, and the forecast is not allowed to use the temperature-0 pass rate. Each arm is predicted from how often it submitted at temperature 0, times the temperature-1.0 pass rate (0.399, 0.431, 0.433, 0.398). For the empty slate that is about $0.61 \times 0.399 = 0.243$; the later measurement was 0.250. If that 40% had been special to temperature 1.0, the four forecasts would have missed. All four landed within 0.03, and the temperature-0 pass rates, measured afterwards, are 0.415, 0.406, 0.417, 0.388, still near 40%. The same pass rate traveled to the other temperature. The slate and the temperature change how often the agent submits; temperature 1.0 resolves higher by 0.010 to 0.032 in every arm because more runs submit. On this slice the examples show a gain on finding the right files. They do not show a reliable gain on how many tasks are solved: resolve moves from 0.260 to 0.320, the uncertainty interval includes no difference, and against BM25 the gap is 0.010.

Airline ablation. Additive text raises refusal and lowers writes. Gold-entity recall is already 1, empty-slate flight recall is 0.90, and full coverage ties the empty slate at $11/20$ official. Leftover write misses are mostly the database hash. Forced gates failed: an origin/destination veto blocked a gold modify; a full-entity allow-list reached fired-id recall 0.78 on writes; a search overlay reached parent write-flight recall 0.14; a reservation-id mask never fired. No filter separated refuse from write with zero write false positives and full refuse recall.

H FACTOID CATALOGS

Section 7 injects claims from two frozen catalogs. A factoid either emits one claim or abstains. Abstention adds nothing to the slate. File and identifier lists are de-duplicated in encounter order. Where a cap is stated, it is the implementation constant. Later factoids may read caches written by earlier ones, so the lists below are application order. Consensus reweighting then selects a subset of the claims that fired; the catalog is the full list, not that subset.

H.1 SWE-BENCH VERIFIED

Fourteen factoids. Sandbox searches use `rg`, falling back to `grep`, with a timeout of 30 seconds. Unless a factoid says otherwise, resolved paths drop tests and are capped at 8.

issue_error_signature. Evidence: issue text. Scan for a traceback header, an `Error`, `Exception`, or `Warning` line, or an assert line. Fire "Issue error signature: " plus the unique matches joined by "; ". Abstain if the issue is empty or the scan misses.

mentioned_paths_symbols. Evidence: issue text. Collect repo-style paths that contain an extension, and CamelCase tokens. Fire "Issue mentions " plus a "paths . . . " part and/or a "symbols . . . " part, joined by "; ". Abstain if both lists are empty. Snake-case symbols are ignored. The checkout is not opened.

test_name_from_issue. Evidence: issue text. Collect `test_*` names and `tests?/` paths. Fire "Issue states tests: " plus the names, then the paths, comma-joined. Abstain if both are empty. The checkout is not opened.

issue_traceback_files. Evidence: issue text. Read `File "..."` frames whose suffix is `py`, `js`, `ts`, `go`, `java`, or `rb`. Strip a `/testbed` prefix, drop test paths, and cap at 8. Fire "Issue traceback files: " plus the comma-separated paths. Abstain if none remain. Frames are cached for later factoids.

symbol_def_sites. Evidence: issue text and the checkout. Abstain if there is no sandbox, no CamelCase symbol, or no hit. For each CamelCase symbol, search `*.py` for `class Symbol` or `def Symbol` until 8 hits. Fire "Symbol definition sites: " plus "symbol at path" items joined by "; ". Python only; test paths are dropped.

import_neighbors. Evidence: issue text and the checkout. Take `*.py` paths named in the issue and search for `import` or `from` lines that name that module stem. Fire "Import neighbors: " plus the unique file paths on those hit lines. Abstain if there is no sandbox, no `*.py` path, or no hit. This factoid has no cap and does not drop test paths.

resolve_partial_paths. Evidence: issue text and the checkout. Extract path fragments that end in a source suffix. For each fragment, list files matching that fragment until 8 hits. Fire "Resolved issue paths: " plus the comma-separated paths. Abstain if there is no sandbox, no fragment, or no file. The listing can over-match. Tests are dropped.

identifier_def_sites. Evidence: issue text and the checkout. Take snake-case and `SCREAMING` identifiers, drop stopwords, prefer names that appear in a call or in backticks, and keep at most 12 names. One batched search looks for `def name` or `name =` in `*.py`. Prefer paths whose basename stem overlaps the name, and cap the claim at 8. Fire "Identifier definition sites: " plus "name at path" items joined by "; ". Abstain if there is no sandbox, no identifier, or no hit. CamelCase names are left to `symbol_def_sites`. Tests are dropped.

error_code_sites. Evidence: issue text and the checkout. Extract Django-style codes of the form `app.E123`. Search each code as a fixed string in `*.py` until 8 hits. Fire "Error code sites: " plus "code at path" items joined by "; ". Abstain if there is no sandbox, no code, or no file. Other code shapes are ignored. Tests are dropped.

test_import_targets. Evidence: issue text and the checkout. Reuse the test names and test paths from `test_name_from_issue`. If the issue names a test but not a path, find a file whose name contains that test. Read `from` and `import` lines at the start of each test file, skip the `tests`, `test`, `unittest`, and `pytest` roots, and resolve the remaining modules to a `.py` file or a package `__init__.py`. Fire "Test import targets: " plus the comma-separated production paths, capped at 8. Abstain if there is no sandbox, no test mention, no test file, or no resolved production file.

issue_snippet_clone. Evidence: issue text and the checkout. Require a markdown fence with at least three non-empty lines. Search `*.py` for the longest distinctive line in the fence (at least 12 characters and containing one of `_(=).`). Score files by how many fence lines they contain, drop tests, and cap at 8. Fire "Issue snippet clone: " plus the comma-separated paths. Abstain if there is no sandbox, no qualifying fence, or no file.

issue_basename_files. Evidence: issue text and the checkout. Keep at most 12 tokens: a lowercase run of at least 8 letters, a snake-case part of at least 8 letters, or a PascalCase word of at least 8 letters that is not CamelCase. List files whose names contain a token. Drop tests and any path component `docs`, `doc`, or `build`. Keep source suffixes. A directory-like hit is expanded to the files under it. Fire "Issue basename files: " plus the comma-separated paths, capped at 8. Abstain if there is no sandbox, no token, or nothing left after the filters.

likely_edit_files. Evidence: the issue and caches from earlier factoids. This factoid does not search the checkout. Prefer paths resolved by `resolve_partial_paths`. Otherwise union, in this order, `issue_traceback_files`, `error_code_sites`, `test_import_targets`, `issue_snippet_clone`, `identifier_def_sites`, `symbol_def_sites`, and `issue_basename_files`. Otherwise use paths written in the issue. Fire "Likely edit files: " plus the comma-separated paths, capped at 8. Abstain if all three sources are empty. The claim is a hypothesis.

fix_strategy. Evidence: issue text. Take the first `Error`, `Exception`, or `Warning` class name, not the rest of the line. Fire "Likely need a guard or type check to stop {Name} from recurring." Abstain if no such class appears. The claim is generic and is not a file localization.

H.2 τ^2 AIRLINE

Sixteen factoids. This is the catalog `load_catalog` registers and the slate builder applies. Identifier lists are capped at 8. The database clock used for booking age is 2024-05-15 15:00:00. Two further factoids are not in this slate; they are the ablations in Appendix H.3.

User identifiers match a lowercase word, an underscore, a lowercase word, an underscore, and four digits. Reservation identifiers are six characters from `A--Z` and `0--9`, excluding flight numbers. Flight numbers match `HAT` plus three digits. Payment identifiers match `credit_card`, `gift_card`, or `certificate`, an underscore, and digits.

user_intent_span. Evidence: the conversation, matched as a lowercase substring. Labels and needles, in list order: cancel (`cancel`, `cancellation`, `refund the booking`); book (`book`, `booking`, `new reservation`); change flights (`change flight`, `change the flight`, `changing`, `modify flight`, `modify reservation`, `reschedule`, `return flight`, `switch to`, `switch the flight`, `different flight`); bags (`bag`, `baggage`, `suitcase`); passengers (`passenger`, `remove`, `change the name`); compensation (`compensat`, `certificate`, `voucher`, `delayed`); insurance (`insurance`); lookup (`how many`, `status`, `details`). Fire "User intent: " plus the unique labels, comma-joined. Abstain if the conversation is empty or no needle hits. Labels are cached.

mentioned_ids. Evidence: the conversation. Extract user, reservation, flight, and payment identifiers, plus a last-four preceded by $\star$ or by `card`. Fire "Mentioned ids: " plus every non-empty list, comma-joined, in that field order. Abstain if every list is empty. The extract is cached.

constraint_keywords. Evidence: the conversation, as a lowercase substring, in this order: `basic economy`, `24 hour`, `24 hrs`, `24 hours`, `insurance`, `gold`, `silver`, `business`, `delayed`, `sick`, `health`, `weather`. Fire "Constraint keywords: " plus the hits, comma-joined. Abstain if none occur. Matching is not restricted to word boundaries.

user_claim_span. Evidence: the conversation. Record a claim when the user says "I'm / I am / we are" plus gold, silver, or regular; "N hours ago"; "I bought / I have insurance" or "I insurance"; or "N passengers". Fire "User claims: " plus those sentences joined by "; ". Abstain if none match.

policy_clause_hits. Evidence: the conversation and the wiki. Split the wiki into sentences, dropping heading lines. Keep a sentence, up to 8, when it contains a policy needle and the conversation contains a trigger for that needle. Needles and triggers: "24 hrs" or "24 hours" with 24-hour wording or cancel; "basic economy flights cannot be modified" with basic economy, modify, or change; "health or weather" with insurance, sick, health, weather, or cancel; "already been flown"

with flown, landed, cancel, or change; "do not proactively offer a compensation" with compensation, certificate, or delayed; "already be in user profile" with payment, card, certificate, or gift; "add but not remove checked bags" with bag, baggage, or suitcase. Fire "Policy clauses: " plus the sentences separated by spaces. Abstain if the conversation or the wiki is empty, or no sentence qualifies.

`reservation_index`. Evidence: the database. Require a user id from the conversation, preferring the cache from `mentioned_ids`. Call `get_user_details` for the first id only. Fire "User {id} reservations: " plus the reservation ids, comma-joined, capped at 8. Abstain if there is no database, no user id, or no reservations. The user record is cached.

`entity_lookup`. Evidence: the database. For each reservation id already mentioned, capped at 8, call `get_reservation_details`. Fire "Entity lookup: " plus, for each record, the id and any `cabin=` and `insurance=` fields, separated by "; ". Abstain if there is no database, no reservation id, no record, or the rendered claim contains a write-tool name (`book_reservation`, `cancel_reservation`, `update_reservation_flights`, `update_reservation_baggages`, `update_reservation_passengers`, `send_certificate`). Records are cached.

`membership_level`. Evidence: the cached user from `reservation_index`, or a fresh `get_user_details` for that user id. Fire "membership={level}". Abstain if there is neither a database nor a cached index, or the record has no membership. The level is cached.

`reservation_cabin_insurance`. Evidence: the `entity_lookup` cache only. For each reservation with a cabin or an insurance flag, emit "{id} cabin={cabin} insurance={flag}", using `unknown` when insurance is missing. Join records with "; ". Abstain if the cache is empty or no record has either field.

`cancel_window_facts`. Evidence: `created_at` on cached reservations, compared with the fixed clock above. Fire "{id} created_at={timestamp} hours_since_booking={hours}" to one decimal, joined by "; ". Abstain if the cache is empty or no record has `created_at`. Hours are cached.

`segment_status_facts`. Evidence: the database and cached reservations. For each flight with a number and a date, call `get_flight_status`. Fire "{number} {date} status={status}", joined by "; ". Abstain if there is no database, the cache is empty, or no segment has both fields. Status strings are cached.

`passenger_count`. Evidence: the passenger list on cached reservations. Fire "{id} passenger_count={n}", joined by "; ". Abstain if the cache is empty. Counts are cached.

`bag_allowance_facts`. Evidence: cached membership and the first cached cabin. Look up free checked bags per passenger in Table 5. Fire "Policy free bags per pax given membership and cabin: {n}". Abstain if membership or cabin is missing, or the pair is not in the table.

Table 5: Free checked bags per passenger. Rows are membership, columns are cabin.

Membership	Basic economy	Economy	Business
regular	0	1	2
silver	1	2	3
gold	2	3	4

`payment_methods_on_file`. Evidence: payment methods on the cached user record. For each method emit its id, source, and last four, when present. Fire "Payment methods on file: " plus those fields, comma-joined. Abstain if the user cache or the method list is empty, if nothing was emitted, or if the claim contains `$` or `amount=`.

likely_write_tools. Evidence: cached intent labels only. No tool call. Map cancel to cancel_reservation, book and insurance to book_reservation, change flights to update_reservation_flights, bags to update_reservation_baggages, passengers to update_reservation_passengers, and compensation to send_certificate. Lookup maps to nothing. Fire "Likely write tools: " plus the unique tool names. Abstain if no label is cached or none of them map to a tool. The claim is a hypothesis.

likely_target_reservations. Evidence: reservation ids already extracted from the conversation, capped at 8. Fire "Likely target reservations: " plus those ids. Abstain if none were named. The claim is a hypothesis.

policy_decision. Evidence: cached reservations, cached hours, cached segment status, cached intent, and the conversation. Abstain if the reservation cache is empty or no predicate fires. Otherwise fire "Policy decision: " plus the unique predicates below, comma-joined. Hours default to 999 when a reservation has no cached age, so a missing age counts as older than 24 hours.

- cancel_eligible=false: age greater than 24 hours, cabin basic economy, insurance no or false, and no segment status landed or flying.
- modify_flights_blocked_basic_economy: cabin contains "basic economy" and the intent is change flights or cancel.
- cabin_change_blocked_flown: a segment status is landed or flying.
- insurance_reason_uncovered: insurance is no or false, or the conversation contains both "birthday" and "insurance".
- compensate_blocked_no_change_or_cancel: the intent includes compensation and includes neither cancel nor change flights.
- od_change_not_allowed: the intent is change flights, the reservation has an origin and a destination, and a conversation airport code, taken from list_all_airports, falls outside that pair.

H.3 Airline ablations

The main slate does not load these factoids. target_flight_search is added only with the stage-S flag, which the search overlay uses. no_legal_write is added only with the write-legal flag. Section 7 reports both as failed forced gates.

target_flight_search. Evidence: the database, the conversation, cached intent, cached reservations, and cached policy predicates. Abstain if there is no database, or if modify_flights_blocked_basic_economy is already cached. Dates are 2024 ISO dates and "May D" phrases in the conversation. The phrases "the day after", "next day", "by one day", and "back by one day" also add the following day. A change-flights intent also adds 2024 dates already on the reservation. When the intent is book or change flights and at least one date exists, search search_direct_flight and search_onestop_flight. Spoken city "New York" expands to JFK, EWR, and LGA. The phrase "west coast" expands destinations to SFO, LAX, SEA, LAS, and PHX. A change uses the reservation route and its segments; a booking uses the spoken airports. Also emit flight numbers already on the reservation, with no search, when the intent is bags or passengers, or the user mentions a cabin change. Fire "Target flight search: " plus the flight numbers separated by spaces. Abstain if the search returns nothing and that reservation fallback does not apply.

no_legal_write. Evidence: cached intent and cached policy predicates. No tool call. Ignore the lookup intent. Map cancel to cancel, book and insurance to book, change flights to modify, bags to bags, passengers to passengers, and compensation to certificate. A predicate forbids a family: cancel_eligible=false forbids cancel; modify_flights_blocked_basic_economy and od_change_not_allowed forbid modify; cabin_change_blocked_flown forbids modify and cancel; compensate_blocked_no_change_or_cancel forbids certificate. The two modify predicates still leave a cancel-and-book bypass when those families are not themselves

forbidden. Fire “No legal write: no_legal_write” only when every requested family is forbidden and no bypass is open. Otherwise abstain.