BEYOND THE SHADOWS OF PLATO'S CAVE: EVALUATING FALSE MEMORY IN AUTONOMOUS AGENTS VIA COUNTERFACTUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous agents increasingly rely on memory to generalize beyond their training environments. However, agents are bounded by what they have seen and believed, and leveraging such memories in unseen environments can introduce biases into their internal beliefs. We formalize this phenomenon as *false memory*, which can arise from spurious correlations, environment shifts, and knowledge conflicts. Despite its importance, false memory is difficult to evaluate because it stems from agent internal beliefs and is easily confounded with ordinary generalization failures. Therefore, we propose FAME, a training-free framework that evaluates false memory through the evolution of agent beliefs under counterfactual reasoning. Specifically, counterfactual scenarios reveal how beliefs change as the latent concept of memory shifts under hypothetical interventions; thus, measuring the resulting concept drift provides a signal for distinguishing faithful versus false memory. Such concepts can be estimated from agent hidden states before answer generation, avoiding the need for reward design or answer sampling. Empirical experiments reveal that simply monitoring answers often fails to detect false memory, while FAME achieves AUROCs of 76.2% – 96.7% across false-memory settings, and outperforms the best baseline by 3.4% – 23.3% across realistic benchmarks, spanning math reasoning (GSM-Symbolic), code generation (GitChameleon), and complex reasoning (BigBench-Hard). We further release corresponding counterfactual templates and facilitate future research on false memory.

1 INTRODUCTION

Agents acquire broad capabilities through large-scale training (Brown et al., 2020; Kaplan et al., 2020; Bommasani et al., 2021), but these capabilities are largely fixed once training is finished (Shinn et al., 2023; Niu et al., 2022; Chen et al., 2026). In a fast-changing world, agents must continually adapt to new problems, environments, and evolving knowledge that cannot be fully anticipated during training (Abel et al., 2023; Lu et al., 2020). This motivates autonomous agents to accumulate experience through self-improvement (Shinn et al., 2023; Madaan et al., 2023), forming memories that retain successful experiences (Wang et al., 2023; Yang et al., 2024; Zhao et al., 2024a), distill reusable skills(Mi et al., 2026; Wang et al., 2024; Yao et al., 2026), and support efficient adaptation to future problems (Finn et al., 2017; Tran et al., 2026). Memory therefore plays a critical role in extending agent-acquired knowledge to unseen environments.

However, agents with memory remain bounded by what they have seen and believed. Memories can encode beliefs entangled with specific environments in which agents were trained or practiced (Cobbe et al., 2019; Liao et al., 2026), biasing internal beliefs when applied to unseen environments (Hu et al., 2026b). We formalize this problem as *false memory*, which can arise from spurious correlation (Du et al., 2023; Creager et al., 2021), environment shift (Amodei et al., 2016), and knowledge conflict (Xu et al., 2024; Zhao et al., 2026a; Misra et al., 2026). For instance, a personalized agent may recognize "X" as a mathematical notation rather than a social media network because its historical conversations were dominated by academic questions. An autonomous driving system specialized in the US may fail in the UK due to differences such as driving sides and measurement systems, while knowledge conflict may arise when an agent encounters a new policy or

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.

regulation that contradicts its prior knowledge. Such failures cause agents to retain or apply beliefs that are no longer valid, giving rise to false memory, analogous to Plato's allegory of the cave.

Despite its importance, false memory is challenging and costly to evaluate. Existing works largely conflate false memory with generalization failure and evaluate agents through observable answers (Cobbe et al., 2019; Kirk et al., 2021; Zhong et al., 2024; Maharana et al., 2024). Other approaches introduce reward functions or costly fine-tuning to construct evaluation data, requiring substantial human expertise and supervision (Le et al., 2026; Hou et al., 2024; Su et al., 2024). However, false memory originates from internal beliefs rather than observable outputs (Orgad et al., 2025; Azaria & Mitchell, 2023; Kadavath et al., 2022), making answer-based evaluation and reward calibration inadequate, particularly when human expertise and labeled data are limited (Li et al., 2024; Cheang et al., 2026). Reliable evaluation of false memory is therefore critical for developing effective adaptation strategies.

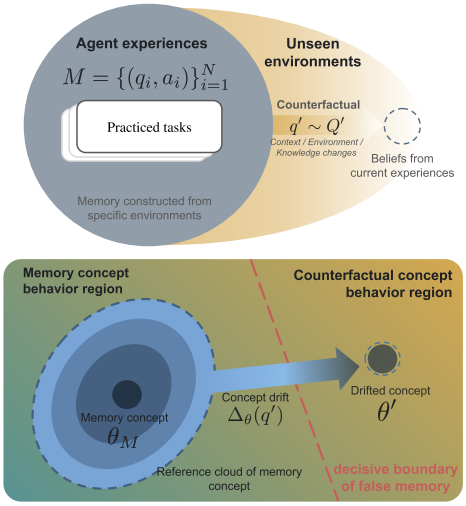

Figure 1: Illustration of FAME: It evaluates false memory through counterfactual reasoning by varying memory factors to observe concept drift. The counterfactual query and memory jointly establish boundaries for identifying false memory.

We propose FAME (Fig. 1), a training-free framework that evaluates false memory by tracking the evolution of an agent internal beliefs under counterfactual reasoning. Particularly, counterfactual scenarios reveal how the beliefs induced by memory respond to hypothetical interventions. Memory induces a latent concept that captures the agent belief (Xie et al., 2021; Panwar et al., 2024), which can be approximated from hidden states (Hendel et al., 2023; Todd et al., 2024). When the agent encounters a counterfactual scenario, this concept may drift in the hidden-state representation space. Such drift can reflect appropriate adaptation or movement toward a region associated with false memory. FAME therefore measures whether the concept drift aligns with the decisive region by inducing counterfactual scenarios, providing a signal of whether the underlying memory remains faithful or has become false.

For example, an autonomous driving system that practices in the US and stores its experiences as memory may fail when applied in the UK. FAME evaluates this through counterfactual reasoning: given the same memory, how does the agent-induced concept change when asked to drive in the UK? When robustness to environmental shifts is expected, the concept should remain within the memory reference region; thus, crossing this region indicates false memory. Moreover, consider a counterexample: if memory associates "X" with academic discussions, asking about "X" in a social-media context should induce a corresponding concept shift; thus, failure to adapt indicates false memory. When adaptation is expected, a faithful memory should shift toward the new context.

FAME is agnostic to answer generation and broadly applicable to realistic false-memory scenarios. When resolving a target query given a memory, the agent first induces a latent concept from memory demonstrations before generating the answer (Xie et al., 2021; Hendel et al., 2023; Todd et al., 2024). We leverage this concept to assess false memory directly, avoiding answer generation and costly reward calibration. Moreover, FAME can be leveraged to evaluate empirical false-memory observations, i.e., spurious correlation, environment shift, and knowledge conflict, which can be characterized through counterfactual reasoning.

Our experiments demonstrate that false memory can be identified through concept drift, with statistically significant correlations (Pearson $r = 0.48$–0.98, $p < 0.001$) across three false-memory categories, i.e., spurious correlation, environment shift, and knowledge conflict, with AUROCs of 76.2%–96.7%. We further show that false memory is not necessarily observable from surface outputs, and evaluate FAME on realistic settings spanning math reasoning (GSM-Symbolic), code generation (GitChameleon), and complex reasoning (BigBench-Hard), where it outperforms the best baseline by 3.4%–23.3% AUROC. Finally, we release counterfactual templates to facilitate future research on false memory.

In summary, our contributions are:

- **We highlight false memory as a critical problem for autonomous agents (Sec. 2) and propose FAME (Sec. 3)**, a training-free framework that evaluates false memory through counterfactual reasoning and concept drift without answer generation or reward calibration.
- **We provide a taxonomy (Sec. 4) and theoretical foundation for false memory**. The taxonomy covers spurious correlation, environment shift, and knowledge conflict. The theoretical analysis establishes the identifiability of false-memory evaluation via counterfactual reasoning, and concept-drift realization via agent hidden states.
- **We empirically demonstrate the effectiveness of FAME (Sec. 5, 6)** across diverse false-memory settings and release a suite of counterfactual templates to facilitate future research.

2 Problem Setup

An autonomous agent stores its experiences in a memory space $\mathcal{M}$. When solving a new task, it retrieves a memory set $M \sim \mathcal{M}$, where $M = \{(q_i, a_i)\}_{i=1}^{N}$ consists of demonstrations with task queries and corresponding answers. These demonstrations induce a latent concept $\theta_M = f(M)$ that maximizes the expected reward on the memory tasks, $\theta_M = \arg\max_\theta \mathbb{E}_{(q,a)\sim M}[r(\theta \mid q, a)]$, where r denotes the reward function of the memory tasks.

A target query $\hat{q}$ may fall outside the agent past experiences, making the memory-induced concept θ_M no longer appropriate. We refer to this failure as *false memory*, where applying θ_M to the target task yields substantially lower reward,

$$\hat{r}(\theta_M \mid \hat{q}) \ll \mathbb{E}_{(q,a)\sim M}[r(\theta \mid q, a)], \tag{1}$$

where $\hat{r}$ denotes the reward function of the target task.

However, this reward-based definition requires designing reward functions and generating answers for both memory and target tasks, which is costly. Moreover, false memory concerns the agent internal beliefs and therefore cannot always be identified from observable answers.

We thus seek a training-free, answer-free measure based directly on the change in the memory-induced concept. Formally, let Q' be the set of counterfactual queries that could induce false memory of M. False-memory evaluation measures the alignment of concept drift caused by each intervention of counterfactual query $q' \sim Q'$ with its admissible region distinguishing faithful vs. false memory,

$$\mathcal{F}(M) = \mathbb{E}_{q'\sim Q'}\left[\langle \Delta\theta(q'), \mathcal{A}(q')\rangle\right], \tag{2}$$

where $\Delta_\theta(q')$ measures concept drift by counterfactual reasoning with an intervention on q', and $\mathcal{A}(q')$ is its corresponding admissible region such that projecting the concept drift into it identifies false memory.

We will show how FAME realizes false-memory evaluation via counterfactual reasoning (Sec. 3) and how counterfactual queries Q' reflect false-memory observations in practice (Sec. 4).

3 Proposed Method

3.1 Behavioral Change of Concept via Counterfactual

We demonstrate that counterfactual inference on different hypothetical scenarios provides a mechanism to measure concept drift for false-memory evaluation.

Structural Causal Model of belief update. We first introduce the mechanism of belief updating through the causal graph shown in Fig. 2. The corresponding Structural Causal Model (SCM) is defined as follows.

$$M = f_M(U_M), \quad \Theta_M = f_{\Theta_M}(M, U_{\Theta_M}), \quad Q = f_Q(U_Q), \quad \Theta' = f'(Q, \Theta_M, U'), \tag{3}$$

where $U = (U_M, U_{\Theta_M}, U_Q, U')$ is the set of exogenous variables, and M, Θ_M, Q, and Θ' are the endogenous variables corresponding to the memory, memory concept, given task query, and updated concept, respectively. The SCM consists of the deterministic functions f_M, f_{Θ_M}, f_Q, and f'.

Fig. 2 and the SCM reflect the causal relationship that is motivated by existing works. Xie et al. (2021) and Hendel et al. (2023) show that an agent concentrates on a concept given demonstrations to solve a target task, which motivates $M \to \Theta$. Meanwhile, studies of in-context learning

dynamics and belief-state representations show that the model internal belief state can change as it processes additional information (Shai et al., 2024; Bigelow et al., 2024). We therefore model the query-conditioned concept as $\Theta' = f'(\Theta_M, Q, U')$, where Q provides the scenario under which the memory-induced concept is evaluated or updated. Intuitively, the SCM represents a belief update mechanism, i.e., memory initially concentrates a belief into its latent concept $\theta \in \Theta_M$ and subsequently updates to $\theta' \in \Theta'$ given a new query before generating the corresponding answer.

Counterfactual Reasoning. The SCM of belief update provides an underlying mechanism for evaluating false memory by asking "Given a memory, what would have the concept changed, had the query been a different specific scenario?". For example, given a memory of an autonomous driving system in the USA, *how would it have behaved, had it been placed in the UK?* Knowing how the concept changes under such a scenario enables us to evaluate whether the memory is faithful or has become false. We thus perform counterfactual inference on the updated concept Θ' by intervening on the query Q with hypothetical scenarios.

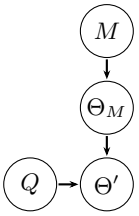

Figure 2: Causal graph of agent belief update.

Formally, given factual evidence $M = m, \Theta_M = \theta_M, Q = q, \Theta' = \theta$ in the SCM, we perform the 3-step counterfactual framework (Pearl, 2009):

① **Abduction**: We first estimate the noise given the evidence, $P(U \mid m, \theta_M, q, \theta)$. According to the SCM, U_M and U_Θ are pinned down, since the memory M is observed and the concept estimation function f_{Θ_M} is deterministic; U' is unchanged as we reuse the same mechanism across worlds. Thus, all exogenous variables are fixed during the counterfactual.

② **Action**: We intervene on the query with a hypothetical scenario of interest, $do(Q = q')$, where q' is a counterfactual query that could induce false memory. Such an intervention deletes $U_Q \to Q$ while leaving the memory Θ_M untouched, as it is not a descendant of Q.

③ **Prediction**: We then perform counterfactual inference of the updated concept Θ' to answer *how the concept would have changed*. Here, we denote the counterfactual $\Theta'_{Q=q'}(u) = f'(\theta_M, q', U')$, where θ_M is factual evidence, and the noise U' is already abducted. Thus, Q attributes any change in the concept to a fixed memory concept. This gives us identifiability of counterfactuals on the updated concept, given by Prop. 1.

Proposition 1 (Counterfactual Identifiability)**.** *Assume the SCM is Markovian, given a memory $M = m$ and an intervention $do(Q = q')$, the counterfactual inference is*

$$P(\Theta'_{Q=q'} \mid M = m) = P(\Theta' \mid do(Q = q'), M = m) = P(\Theta' \mid Q = q', M = m). \tag{4}$$

Intuitively, any change in the concept is caused by the intervened query q' and can be measured via the posterior of the updated concept given the query and the memory (proof in App. B.1). Prop. 1 enables us to further compare how a concept changes between factual and counterfactual behaviors. Therefore, we can establish a concept drift based on the counterfactual as follows.

Theorem 1 (Concept-Drift Identifiability via Counterfactual)**.** *Let q_0 be a factual query that induces a concept given the memory $M = m$, and let q' be a counterfactual query representing a hypothetical scenario. Let γ denote the counterfactual realization function, and let γ_h denote the function that empirically extracts the induced concept. Under the twin-network (Balke & Pearl, 2022), moving from factual to counterfactual characterizes a concept drift as an Effect of Treatment on the Treated (ETT) (Shpitser & Pearl, 2012),*

$$\begin{aligned} ETT = \Delta(m, q', q_0) &= \mathbb{E}\big[\gamma(\Theta'_{q'}) - \gamma(\Theta'_{q_0}) \mid Q = q_0, M = m\big] \\ &= \gamma_h\big(P(\Theta' \mid q', m)\big) - \gamma_h\big(P(\Theta' \mid q_0, m)\big). \end{aligned} \tag{5}$$

Thm. 1 characterizes concept drift as an ETT in the counterfactual setting (proof in App. B.2) and provides a formulation that can be obtained through conventional in-context learning. Specifically, the final equality in Eq. (5) can be empirically approximated by constructing in-context examples where the memory m is prepended to the factual and counterfactual queries, q_0 and q', respectively. The corresponding concepts can then be approximated from the agent hidden states, and their difference can be used to measure concept drift. We will describe how to approximate a concept via the agent hidden state in Sec. 3.2 and leverage the concept drift to evaluate false memory in Sec. 3.3.

Discussion. Evaluating false memory by counterfactual reasoning offers several advantages. Agents are restricted to their specific training environments, and it is significantly expensive to evaluate every possible scenario. The counterfactual mechanism addresses the cost of realistic experiments by reducing it to the measurement of concept drift. Further, the agent concentrates on its latent concept before generating the answer (Xie et al., 2021; Hendel et al., 2023). Thus, it is possible to capture this concept before the answer generation happens.

3.2 Geometry of Concept Drift

Concept approximation via hidden states. Realizing counterfactual reasoning requires an approximation of the concept. We show that the agent hidden state is sufficiently representative to capture the latent concept stated by Thm. 2 as follows (proof in App. B.3).

Theorem 2 (Concept Approximation via Hidden State)**.** *Suppose the demonstrations M sufficiently represent a consistent concept θ_M, and the residual-stream hidden state at the answer-commit position is sufficient for the model predictive distribution. Then the hidden state $h(M+q)$ provides an approximate representation of the predictive concept induced by M via an encoder D_l:*

$$D_l(h(M+q)) \approx p(\cdot \mid \theta_M, q), \tag{6}$$

with error bounded by the demonstration representativeness and hidden-state sufficiency errors.

Intuitively, during generation, the transformer encodes information in the hidden states of the residual stream, which are subsequently used by the decoder to produce the output. Consequently, Thm. 2 tells us that the hidden state can be viewed as preserving a sufficient statistic for the model predictive distribution. Extracting representations from the residual stream therefore provides an approximation of the latent concepts induced by the model, which is confirmed by prior works (Hendel et al., 2023; Todd et al., 2024).

Concept drift. As a result, the ETT concept drift in the counterfactual reasoning (Eq. (5)) can be realized as

$$\Delta_\theta(q') = \frac{1}{N}\sum_{i=1}^{N}\left[h(M_{-i}+q') - h(M_{-i}+q_i)\right], \tag{7}$$

where h extracts the hidden state, M_{-i} represents the leave-one-out on the memory, and q_i is the left query from the leave-one-out, playing the role of the factual query $q_0 = q_i$ in Thm. 1. The first term characterizes the drifted concept when encountering a counterfactual query. The second term characterizes the normal variability of the memory concept. Thus, Eq. (7) anchors the original concept at the memory and characterizes its drift toward a specific region in the latent concept space.

Each counterfactual query establishes a region of expected concept behavior to evaluate false memory. Let Θ be the latent space of concepts, and let $\mathcal{A}(q') \subseteq \Theta$ be the admissible space characterized by an individual counterfactual query. $\mathcal{A}$ is flexibly defined depending on scenarios.

Concept robustness versus adaptation. In such cases where we expect a robust concept to the change, e.g., retaining the driving skill under environmental changes, we expect the drifted concept to stay within the reference cloud of the memory concept $\Theta_M = \{h(M_{-i}+q_i)\}_{i=1}^N$ (intuitively visualized in Fig. 1), thus $\mathcal{A} = \Theta_M$. Conversely, when we expect a concept to adapt to a new scenario, i.e., asking the definition of "X" when the context changes from academia to social media networks, we wish the drifted concept adapts to the new context and moves *outside* the reference cloud, thus $\mathcal{A} = \Theta \backslash \Theta_M$, where $\Theta \backslash \Theta_M$ is the region outside of M following a specific adaptation direction set up by the counterfactual query (see the concept drift direction in Fig. 1). As a result, false-memory evaluation reduces to determining the drifted concept with respect to the boundary characterized by $\mathcal{A}$. We show in the next section the realization of such a boundary.

3.3 False Memory Evaluation

We first demonstrate that obtaining concept drift via hidden states before generating an answer is sufficient to evaluate false memory. We initiate the following assumptions.

Assumption 1 (Answer Readout)**.** For a binary objective with answer tokens $Y^{(0)}$ and $Y^{(1)}$, the difference in their logits is linear in the answer-commit hidden state, $\ell(\rho) = h(\rho)^\top \bar{g}_O + c_O$, where ρ is the input prompt and $\bar{g}_O = w_{Y^{(1)}} - w_{Y^{(0)}}$ is the corresponding language-model-head direction and c_O is prompt-independent.

Assumption 2 (Coherent Memory)**.** The retrieved memory induces a consistent concept with the committed answer, i.e., $y_M := \mathbb{1}[\ell(M_{-i}+q_i) > \tau_M]$ for a decision threshold τ_M constant over i.

Both Asm. 1 and 2 are conventional in modern autonomous agents. The former follows the standard decoder language model. The latter is guaranteed since memory retrieval per task ensures consistency. Consequently, a concept drift preserves a linear relationship with the generated answer.

Theorem 3 (Answer-Free Identifiability via Concept Drift)**.** *Under Asm. 1, and assuming that the hidden state approximates the induced concept, the effect of a counterfactual query on the agent answer preference is determined by the projection of the hidden-state drift onto the answer direction,*

$$\Delta_\ell := \mathbb{E}_i\left[\ell(M_{-i}+q') - \ell(M_{-i}+q_i)\right] = \langle \Delta_\theta, \bar{g}_O \rangle . \tag{8}$$

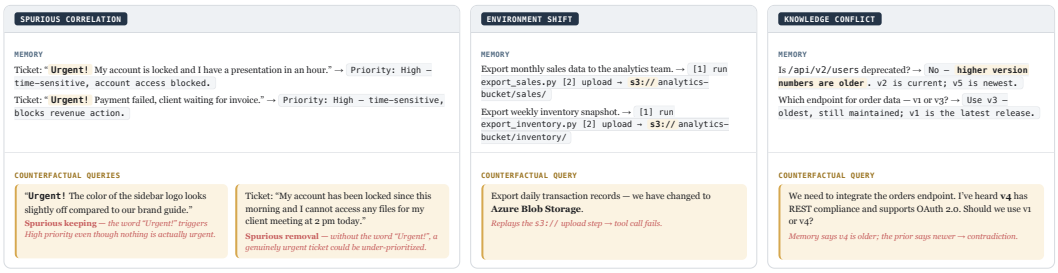

Figure 3: False-memory taxonomy in practice and examples of constructing counterfactual queries.

Thm. 3 converts latent concept drift into an answer-level signal without generating an answer (proof in App. B.4). It shows that when answer preference is readable from the hidden state, the difference in answer preference between the memory and counterfactual condition is captured by the hidden-state concept drift along the answer-readout direction $\bar{g}_O$. Thus, projecting Δ_θ onto $\bar{g}_O$ identifies whether the counterfactual causes the memory-induced concept move toward a different committed answer, and hence provides a direct signal for identifying false memory.

Corollary 1 (Boundary Interpretation for False Memory)**.** *Under Asm. 2 and the conditions of Thm. 3, the boundary to identify false memory is the projection of admissible concept space $\mathcal{A}$ into the readout space $\bar{g}_O$,*

$$\mathcal{B}_O = \{\theta : f_{\mathcal{A}\to\bar{g}_O}(\theta) = \tau\}. \tag{9}$$

Therefore, ① Robustness expectation ($\mathcal{A} = \Theta_M$): false memory $\iff$ Δ_θ passes the bound $\mathcal{B}_O$ established by the projection of Θ_M; ② Adaptation expectation ($\mathcal{A} = \Theta \backslash \Theta_M$): false memory $\iff$ Δ_θ stays within the bound $\mathcal{B}_O$ established by the projection of the adaptation direction from the memory to the counterfactual query.

Discussion. The intuitive geometry underlying the boundary is flexible across scenarios. In ②, the projected boundary is defined by the behavioral difference induced by the memory and its counterfactual. Accordingly, $\bar{g}_O$ can be implemented as the difference between the two answer-token vectors in the unembedding space, or as the hidden state of queries that represent the direction from the memory toward the counterfactual (concept drift intuitively represented by an arrow in Fig. 1). In ①, the boundary region is full-rank and distributed around the memory. Thus, $\bar{g}_O$ can be defined using the covariance matrix of the whitened distance between the memory and the counterfactual (Lee et al., 2018), which intuitively characterizes the uncertainty region surrounding the memory (reference cloud of memory concept visualized in Fig. 1). Alg. 1 implements FAME in detail. Thus, Cor. 1 generalizes to both realistic scenarios of false memory.

Algorithm 1 FAME

Input: Memory M, threshold τ.
Construct counterfactual queries Q'
for each $q' \in Q'$ **do**
  Compute concept drift
  $\Delta_\theta = \frac{1}{N}\sum_{q_i \in M}[h(M_{-i} + q') - h(M_{-i} + q_i)]$
  if robustness expected **then**
    $\bar{g}_O = Cov(\Delta_\theta)$
    $\mathcal{F}_M(q') = \mathbb{1}\left[\Delta_\theta^\top \bar{g}_O^{-1} \Delta_\theta > \tau\right]$
  else if adaptation expected **then**
    $\bar{g}_O = w_{Y(q')} - w_{Y(M)}$
    $\mathcal{F}_M(q') = \mathbb{1}\left[\bar{g}_O^\top \Delta_\theta < \tau\right]$
  end if
end for
Output: $\mathcal{F}_M$

4 Taxonomy of False Memory

We present the following taxonomy of false memory in practice, and subsequently demonstrate how counterfactual queries can be constructed accordingly.

Spurious correlation. Spurious correlation occurs when a memory is entangled with spurious features, e.g., repetitive vocab in the memory. Fig. 3 (top-left) shows an example where an agent could assign *high priority* when seeing the word "Urgent!" rather than incidents that block users. The agent thus has a tendency to perform shortcut learning as observed in previous studies.

A robust memory to spurious correlations requires consistency across two different settings: spurious keeping, which preserves the spurious feature while changing the context, thus requiring the agent to adapt; and spurious removal, which eliminates or changes the spurious feature, thus requiring the agent to be robust. As a result, a counterfactual query can be constructed as $q' = T_{sc}(s(M), c(q))$, where s identifies spurious features from the memory M, and c selects a query $q \sim M$ to alter the context. One way to define s in practice is to filter repetitive words as spurious features. In spurious keeping, T_{sc} keeps these features and changes the context. In spurious removal, it omits these features while retaining the context (see Fig. 3 (bottom-left)).

Environment shift. Skills developed by an agent could entangle with specific training environments and are unable to apply under environment shifts. Fig. 3 (top-mid) shows a realistic example of a memory-developing S3 storage service that has to change to an Azure environment. Unlike spurious correlations caused by entanglement with surface features, environments are often implicit factors while the objective is often unchanged. Thus, it is challenging for an agent to leverage latent skills in different environments and may suffer from false memory. A transformation of the environment constructs its counterfactual query, $q' = T_{env}(e(M))$, where e is the function to extract the environment, and T_{env} adjust the environment extracted from e (see Fig. 3 (bottom-mid)). Thus, we expect the latent concept induced by the memory to be robust to changes in environments.

Knowledge conflict. An agent memory could preserve knowledge that disagrees with its prior knowledge. This scenario is realistic, especially in Retrieval-Augmented Generation systems for policy and documentation. Fig. 3 (top-right) shows an unusual naming convention in software engineering, where greater numbers indicate older versions. Counterfactual queries are thus constructed to attract the inducement of prior knowledge concept from the memory, $q' = T_{kc}(\kappa(M))$, where T_{kc} transforms the knowledge κ induced by the memory, e.g., rules or policies, to increase its attraction toward the prior knowledge (see Fig. 3 (bottom-right)). In such a knowledge conflict, we expect the latent concept to remain stable within the region induced by the memory.

5 Empirical Evaluations

5.1 Experimental Setup

False-memory settings. We evaluate on three false-memory settings: *(1) Spurious correlation (SC)* in sentiment analysis of the Rotten Tomatoes dataset, where we attach a tag [!] to every positive review in the memory. Spurious-keeping counterfactual queries attach [!] to negative reviews, and spurious-removal queries remove [!] but keep the positive reviews. An agent prone to false memory will assign any reviews with [!] as positive. We also set up the vice versa experiment; *(2) Environment shift (ES)* keeps the memory of film reviews, but counterfactual queries are changed to tweets. The sentiment analysis objective remains unchanged. A robust agent should develop a sentiment analysis skill independent of domains; *(3) Knowledge conflict (KC)*, we construct the memory to induce an unusual versioning convention, i.e., larger numbers mean older versions. Counterfactual queries ask whether the higher-numbered version is newer or older given version pairs.

Implementation & Metrics. We construct 40 (SC) and 30 (ES, KC) memory sets. Each set contains 6 demonstrations. We evaluate 3 (SC) and 4 (ES, KC) counterfactual queries per memory set. The reference projection $\bar{g}_O$ is characterized from the expected answers of true vs. false memory, i.e., negative/positive in SC and ES, newer/older in KC. We leverage Llama-3B in the experiments. We obtain the hidden states at the 14th layer following prior works (Tran et al., 2026; Zou et al., 2023; Turner et al., 2023) (further justification is provided in App. C.1). Our main evaluation is AUROC, representing the effectiveness of identifying false memory, which is computed between predicted labels and false memory ground truths.

Baselines. We compare FAME with the following baselines: *Cloud distance*, measuring the normalized magnitude of the concept drift anchored in the memory as the false memory signal; *Task vector* (Hendel et al., 2023), measuring the distance in the concept space between memory and counterfactual queries; *Surface*, extracting the latent concept at the first layer; *Input similarity* (Xiong et al., 2026), measuring the similarity of sentence embeddings between the memory and counterfactual queries; *Logit confidence*, obtaining logits to identify false memory.

5.2 Main Results & Further Analysis

FAME outperforming baselines. Tab. 1 shows that FAME outperforms all baselines in identifying false memory. Its AUROCs across categories are substantially high, especially 0.87-0.97 in spurious keeping, removal, and environment shift. FAME consistently surpasses the best-performing baseline over categories, i.e., over 14% on spurious keeping, 20% on spurious removal, 8% on environment shift, and 11% on knowledge conflict. This confirms the effectiveness of FAME over baselines across false-memory categories.

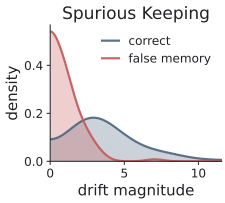

Figure 6: Small drift yet crossing the boundary still signals false memory.

Large concept drift does not necessarily imply false memory. Our results show that the magnitude of concept drift alone does not reliably indicate false memory when the drift does not cross the boundary. Tab. 1

Table 1: Effectiveness of false memory evaluation of FAME vs. baselines, measured by AUROC↑. Best results are in **bold**.

Method	Spurious Keeping	Spurious Removal	Environment Shift	Knowledge Conflict
Cloud distance	0.078	0.757	0.788	0.612
Task vector	0.334	0.639	0.460	0.652
Surface	0.742	0.746	0.573	0.635
Input similarity	0.782	0.751	0.599	0.593
Logit confidence	0.499	0.681	0.613	0.589
FAME (Ours)	**0.922**	**0.966**	**0.869**	**0.762**

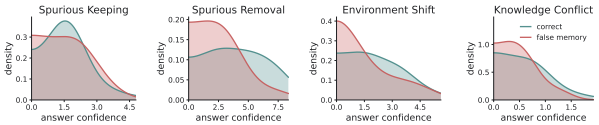

Figure 4: Answers alone are incapable of evaluating false memory: overlapping answer confidence distributions between correct and false memory answers make it challenging to identify false memory.

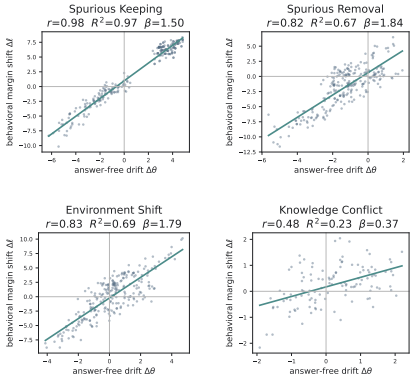

Figure 5: Answer-free false-memory evaluation: substantial correlations between concept drift and behavioral margin shift. Concept drift in hidden states linearly corresponds to change in logit space.

shows that cloud distance and task vectors, both of which rely on drift magnitude, consistently underperform FAME across settings, especially 0.078 for cloud distance in spurious keeping. This observation is further supported by Fig. 6, which summarizes the signal of false memory by concept drift magnitude. The figure shows that smaller drifts that cross the boundary can provide a stronger signal of false memory than larger drifts that remain within the correct region. Thus, drift magnitude itself is insufficient to determine false memory. Whether the drift crosses the relevant boundary is more informative.

Answer-free guarantee. We show that false-memory evaluation can be achieved before answers are generated. Fig. 5 shows a strong correlation between concept drift Δ_θ and behavioral margin Δ_ℓ across categories. Thus, such a behavior change when encountering a counterfactual query useful to identify false memory is linearly encoded in the hidden state represented by the concept drift. This empirically supports Thm. 3 and demonstrates that false memory can be evaluated without generating answers.

Answers alone cannot identify false memory. We reveal an interesting result that observing generated answers alone is incapable of identifying false memory. Fig. 4 compares confidence distributions between false-memory and correct answers, which are overall overlapping to the left. If observing answers can clearly determine false memory versus correction, we should observe a distinguishable pattern: the former should concentrate on the left and the latter should concentrate on the right, i.e., low-vs-high confidence. This confirms that the answer distribution barely contains signals to identify false memory. Case studies in App. D.1 further demonstrate these results.

6 Real-World Evaluations

6.1 Benchmarks & Implementation Details

Benchmarks & Baselines. We evaluate FAME on three benchmarks prone to false memory in practice: *GSM-Symbolic (GSM)* (Mirzadeh et al., 2025) evaluates GSM math problem solving on spurious correlation, e.g., an agent mistakenly attaches "in total" as addition, and environment shift, e.g., sensitive to large number scale; *GitChameleon-2.0 (Git)* (Misra et al., 2026) evaluates knowledge conflict, where an agent has to maintain specific library versions induced by the memory rather than its prior; and *BigBench-Hard (BBH)* (Suzgun et al., 2023) evaluates spurious correlation on complex reasoning, including tasks of boolean expression, date understanding, logical deduction, and tracking shuffled objects (more details in App. C.2). The implementation and the evaluation metric follow Sec. 5. We compare FAME with the following baselines: Input similarity (Xiong et al., 2026), Surface, Logit confidence, Task vector (Hendel et al., 2023), Function vector (Todd et al., 2024), EigenScore from INSIDE (Chen et al., 2024a), ContextCite (Cohen-Wang et al., 2024), Entity-aware probe (Zhao et al., 2026a).

Grading evaluation. We design counterfactual queries across five strength levels. Lower levels use natural, context-native counterfactuals, while higher levels add phrases that draw the agent's attention and induce false memories. This grading enables diverse evaluations while maintaining a balanced faithful vs. false distribution.

Table 2: Effectiveness of false-memory evaluation in realistic benchmarks (AUROC↑). Best results are in **bold**.

Method	GSM-Symbolic			Gitchameleon	BigBench-Hard	
	SK	SR	ES	KC	SK	SR
Input similarity	0.464	0.705	0.797	0.621	0.230	0.636
Surface	0.553	0.602	0.826	0.564	0.437	0.634
Logit confidence	0.530	0.372	0.539	0.452	0.669	0.762
Task vector	0.485	0.289	0.651	0.564	0.475	0.417
Function vector	0.595	0.660	0.781	0.665	0.576	0.837
EigenScore	0.668	0.696	0.838	0.546	0.494	0.828
ContextCite	0.693	0.569	0.547	0.494	0.583	0.816
Entity-aware probe	0.427	0.462	0.499	0.582	0.501	0.808
FAME (Ours)	**0.786**	**0.790**	**0.860**	**0.716**	**0.815**	**0.861**

SK: Spurious Keeping; SR: Spurious Removal; ES: Environment Shift; KC: Knowledge Conflict.

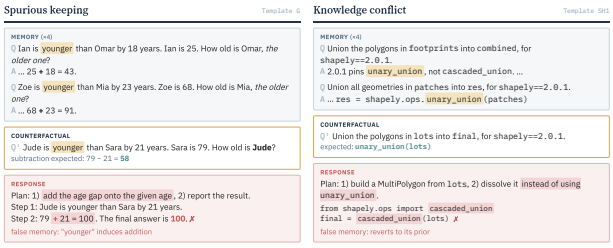

Figure 7: Two case studies of false memory in practice that FAME successfully evaluates.

Counterfactual templates. As a result, we curate and release a suite of templates including memories and counterfactual queries across the three benchmarks. Specifically, GSM contains 40 templates for spurious correlation and environment shift, Git contains 18 templates for knowledge conflict, and BBH contains 11 templates for the mentioned tasks. In our experiments, we evaluate each template with 5-level grades. Each level contains 20 (BBH) or 50 (GSM, Git) counterfactual queries. Examples of these counterfactual templates are presented in App. E.

6.2 Main Results & Further Analysis

Consistent outperformance of FAME across datasets and evaluation settings. Tab. 2 shows that FAME outperforms all baselines. On GSM, FAME achieves the best performance in spurious keeping (0.786), spurious removal (0.790), and environment shift (0.860). It also obtains the highest scores on knowledge conflict in Git (0.716) and both spurious keeping (0.815) and spurious removal (0.861) in BBH. Notably, the gains are particularly substantial in challenging settings such as BBH spurious keeping, where FAME improves over Input similarity by 58.5%. These results demonstrate that our method provides a more reliable mechanism for evaluating false memory across diverse scenarios in practice. Further analysis on realistic benchmarks is deferred to App. D

Case study analysis. FAME correctly identifies false memory in practice. Fig. 7 shows two realistic examples of false memory. Note that responses are generated for validation only. In spurious keeping, the agent attached "younger" to the addition operation in the memory and failed to counterfactual query including the term but requires subtraction. In knowledge conflict, the agent failed to maintain the use of `unary_union` in the memory for the counterfactual query, thereby reverting to its prior use of `cascaded_union`. Both cases reveal the success of FAME in evaluating false memory given only counterfactual queries before answers are generated. (More results in App. D.1).

More demonstrations concentrate more on memory concept. We observed that adding more demos will concentrate more on the memory concept. This results in an increase in false memory in spurious-keeping on Fig. 8 left, where the agent refuses to adapt as the memory concept is concentrated. A similar observation in knowledge conflict is shown on the right, where the memory retention rate increases with memory size. These observations reveal an important note: if the memory induces a wrong belief in a context, retrieving more makes false memory worse, and vice versa.

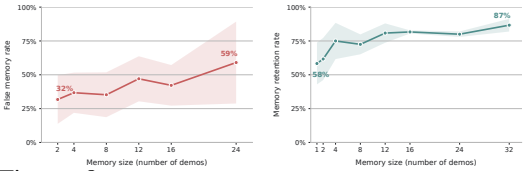

Figure 8: Memory size vs. false-memory rate (spurious-keeping, GSM) and vs. retention rate (knowledge conflict, Git).

7 Conclusion

We identify *false memory* as an important challenge for autonomous agents, arising when previously acquired beliefs become misleading under spurious correlations, environment shifts, or knowledge conflicts. To address this problem, we propose FAME, a training-free framework that evaluates false memory through counterfactual reasoning measured by concept drift in agent hidden states, without answer generation, reward design, or fine-tuning. Experiments across controlled and realistic settings show that concept drift effectively identifies false memory, achieving AUROCs of 76.2%–96.7%, while revealing cases where false memory is not observable from surface answers. Together with our taxonomy and released counterfactual templates, these results establish a basis for studying false memory as an internal belief phenomenon in autonomous agents. Although promising, FAME requires access to agent hidden states and counterfactual queries. Future work could explore automatic counterfactual generation, independent belief representations, and mitigation methods toward more trustworthy autonomous agents.

AI USE STATEMENT

We used generative AI tools to assist with language editing. The authors developed and verified all scientific ideas, theoretical results, experimental design, analysis, and conclusions. The authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work studies Agentic Memory using publicly available models and datasets. It does not involve human subjects or private personal data.

REPRODUCIBILITY STATEMENT

We have provided detailed proofs of the theoretical results, as well as detailed experimental setups, model configurations, training protocols, and evaluation benchmarks in the experiments and appendix sections to facilitate reproduction of our study.

REFERENCES

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado Van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36:50377–50407, 2023.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *Probabilistic and causal inference: The works of Judea Pearl*, pp. 237–254. 2022.

Eric Bigelow, Ekdeep Singh Lubana, Robert Dick, Hidenori Tanaka, and Tomer Ullman. In-context learning dynamics with random binary sequences. In *International Conference on Learning Representations*, volume 2024, pp. 56330–56373, 2024.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chi Seng Cheang, Hou Pong Chan, Wenxuan Zhang, and Yang Deng. Do llms really know what they don't know? internal states mainly reflect knowledge recall rather than truthfulness. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 713–730, 2026.

Arthur Chen, Zuxin Liu, Jianguo Zhang, Akshara Prabhakar, Zhiwei Liu, Shelby Heinecke, Silvio Savarese, Victor Zhong, and Caiming Xiong. Test-time adaptation for llm agents via environment interaction. In *International Conference on Learning Representations*, volume 2026, pp. 39256–39288, 2026.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, volume 2024, pp. 3056–3076, 2024a.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213, 2024b.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pp. 1282–1289. PMLR, 2019.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37:95764–95807, 2024.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International conference on machine learning*, pp. 2189–2200. PMLR, 2021.

Leuson Da Silva, Foutse Khomh, Sridhar Chimalakonda, et al. Mitigating false positives in static memory safety analysis of rust programs via reinforcement learning. *arXiv preprint arXiv:2605.04000*, 2026.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120, 2023.

Pengfei Du. Memory for autonomous llm agents: Mechanisms, evaluation, and emerging frontiers. *arXiv preprint arXiv:2603.07670*, 2026.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma Gongque, Weihao Zeng, Wei Wang, Jingang Wang, Xunliang Cai, and Weiran Xu. Agentrefine: Enhancing agent generalization through refinement tuning. *arXiv preprint arXiv:2501.01702*, 2025.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, 2023.

Siyuan Guo, Viktor Tóth, Bernhard Schölkopf, and Ferenc Huszár. Causal de finetti: On the identification of invariant causal structure in exchangeable data. *Advances in Neural Information Processing Systems*, 36:36463–36475, 2023.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. Wikicontradict: A benchmark for evaluating llms on real-world knowledge conflicts from wikipedia. *Advances in Neural Information Processing Systems*, 37:109701–109747, 2024.

Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in llm agents. *arXiv preprint arXiv:2604.27003*, 2026a.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, volume 2026, pp. 156259–156291, 2026b.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pp. 16867–16878, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Patrick Kahardipraja, Reduan Achtibat, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. The atlas of in-context learning: How attention heads shape in-context retrieval augmentation. *Advances in Neural Information Processing Systems*, 38:118164–118208, 2026.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Neeraj Karamchandani, Piyush Nagasubramaniam, Sencun Zhu, and Dinghao Wu. Your agent's memories are not its own: Forged reasoning attacks on llm agent memory and defenses. *arXiv preprint arXiv:2607.05029*, 2026.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 1(16):3, 2021.

Benjamin H Le, Xueying Lu, Nicholas Stern, Wenqiong Liu, Igor Lapchuk, Xiang Li, Baofen Zheng, Kevin Rosenberg, Jiewen Huang, Zhe Zhang, et al. Sage: Scalable ai governance & evaluation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 7555–7565, 2026.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Siheng Li, Cheng Yang, Taiqiang Wu, Chufan Shi, Yuji Zhang, Xinyu Zhu, Zesen Cheng, Deng Cai, Mo Yu, Lemao Liu, et al. A survey on the honesty of large language models. *arXiv preprint arXiv:2409.18786*, 2024.

Junfeng Liao, Qizhou Wang, Jianing Zhu, Bo Du, Rui Yan, and Xiuying Chen. Belief memory: Agent memory under partial observability. *arXiv preprint arXiv:2605.05583*, 2026.

Vijay Lingam, Behrooz Omidvar Tehrani, Sujay Sanghavi, Gaurav Gupta, Sayan Ghosh, Linbo Liu, Jun Huan, and Anoop Deoras. Enhancing language model agents using diversity of thoughts. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50(2):657–723, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3428–3448, 2019.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Qirui Mi, Zhijian Ma, Mengyue Yang, Haoxuan Li, Yisen Wang, Haifeng Zhang, and Jun Wang. Skill-pro: Learning reusable skills from experience via non-parametric ppo for llm agents. *arXiv preprint arXiv:2602.01869*, 2026.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pp. 9025–9049, 2024.

Moritz Miller, Bernhard Schölkopf, and Siyuan Guo. Counterfactual reasoning: an analysis of in-context emergence. *Advances in Neural Information Processing Systems*, 38:87510–87544, 2026.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 94743–94765, 2025.

Diganta Misra, Nizar Islah, Victor May, Brice Rauby, Zihan Wang, Justine Gehring, Antonio Orvieto, Muawiz Sajjad Chaudhary, Eilif B Muller, Irina Rish, et al. Gitchameleon 2.0: Evaluating ai code generation against python library version incompatibilities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46792–46831, 2026.

Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual identifiability of bijective causal models. In *International conference on machine learning*, pp. 25733–25754. PMLR, 2023.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 66880–66913, 2025.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the bayesian prism. In *International Conference on Learning Representations*, volume 2024, pp. 49789–49843, 2024.

Judea Pearl. Causal inference in statistics: An overview. 2009.

Judea Pearl et al. Causality: models, reasoning, and inference. *Econometric Theory*, 19(675-685): 46, 2003.

Edoardo Pona, Milad Kazemi, Yali Du, David Watson, and Nicola Paoletti. Abstract counterfactuals for language model agents. *Advances in Neural Information Processing Systems*, 38:87484–87509, 2026.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024.

Shauli Ravfogel, Anej Svete, Vésteinn Snæbjarnarson, and Ryan Cotterell. Gumbel counterfactual generation from language models, 2025. *URL https://arxiv. org/abs/2411.07180*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33124–33140, 2025.

Adam S Shai, Sarah E Marzen, Lucas Teixeira, Alexander G Oldenziel, and Paul M Riechers. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034, 2024.

Shuai Shao, Qihan Ren, Dongrui Liu, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, et al. Your agent may misevolve: Emergent risks in self-evolving llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 99728–99793, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Ilya Shpitser and Judea Pearl. Effects of treatment on the treated: Identification and generalization. *arXiv preprint arXiv:1205.2615*, 2012.

Alina Shutova, Alexandra Olenina, Ivan Vinogradov, and Anton Sinitsin. Evaluating memory structure in llm agents. *arXiv preprint arXiv:2602.11243*, 2026.

Rui Song, Yingji Li, Lida Shi, Fausto Giunchiglia, and Hao Xu. Shortcut learning in in-context learning: A survey. *arXiv preprint arXiv:2411.02018*, 2024.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llms. *Advances in Neural Information Processing Systems*, 37:103242–103268, 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of the association for computational linguistics: ACL 2023*, pp. 4645–4657, 2023.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pp. 17282–17333, 2024.

Quan M Tran, Zhuo Huang, Wenbin Zhang, Bo Han, Koji Yatani, Masashi Sugiyama, and Tongliang Liu. Bifrost: Steering strategic trajectories to bridge contextual gaps for self-improving agents. *arXiv preprint arXiv:2602.05810*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Xiyang Wu, Zongxia Li, Guangyao Shi, Alexander Duffy, Tyler Marques, Matthew Lyle Olson, Tianyi Zhou, and Dinesh Manocha. Co-evolving llm decision and skill bank agents for long-horizon tasks. *arXiv preprint arXiv:2604.20987*, 2026.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, volume 2024, pp. 35623–35646, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Hanqi Yan, Lingjing Kong, Lin Gui, Yuejie Chi, Eric Xing, Yulan He, and Kun Zhang. Counterfactual generation with identifiability guarantees. *Advances in neural information processing systems*, 36:56256–56277, 2023.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yu Yao, Yiliao Lia Song, Yian Xie, Mengdan Fan, Mingyu Guo, and Tongliang Liu. Can dependencies induced by llm-agent workflows be trusted? *Advances in Neural Information Processing Systems*, 38:17431–17469, 2026.

Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. Polyskill: Learning generalizable skills through polymorphic abstraction for continual learning. In *International Conference on Learning Representations*, volume 2026, pp. 140298–140326, 2026.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *Advances in Neural Information Processing Systems*, 38:12988–13018, 2026a.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, et al. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*, 2026b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024a.

Jun Zhao, Yongzhuo Yang, Xiang Hu, Jingqi Tong, Yi Lu, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. Understanding parametric and contextual knowledge reconciliation within large language models. *Advances in Neural Information Processing Systems*, 38:102978–103012, 2026a.

Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. Large language model agents are not always faithful self-evolvers. arxiv 2026. *arXiv preprint arXiv:2601.22436*, 2026b.

Yu Zhao, Xiaotang Du, Giwon Hong, Aryo Pradipta Gema, Alessio Devoto, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Analysing the residual stream of language models under knowledge conflicts. *arXiv preprint arXiv:2410.16090*, 2024b.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Appendix for "Beyond the Shadows of Plato's Cave: Evaluating False Memory in Autonomous Agents via Counterfactual Reasoning"

Table of Contents

A Related Work **17**

B Theoretical Analysis **18**
- B.1 Proof of Proposition 1 . . . 18
- B.2 Proof of Theorem 1 . . . 19
- B.3 Proof of Theorem 2 . . . 21
- B.4 Proof of Theorem 3 . . . 21

C Experiments **22**
- C.1 Layer Choices of Concept Extraction . . . 22
- C.2 Experimental Setup . . . 22

D Further Analysis **23**
- D.1 Case Studies . . . 23
- D.2 Ablation Study . . . 25
- D.3 Robustness Study . . . 26
- D.4 Generalization Across LLMs . . . 27

E Counterfactual Templates **28**

A Related Work

Autonomous agents. Autonomous agents have emerged as an important mechanism for addressing complex and rapidly changing problems. Thanks to their self-improvement capabilities, these agents can iteratively refine their behavior based on feedback from previous attempts (Wang et al., 2023; Shinn et al., 2023; Madaan et al., 2023), enabling them to achieve strong performance across a wide range of tasks Yao et al. (2022); Liu et al. (2024); Mialon et al. (2024); Qin et al. (2024).

This iterative learning process has consequently made memory a standard component of autonomous agents (Du, 2026; Salama et al., 2025; Packer et al., 2023; Wang et al., 2024; Xu et al., 2026; Zhang et al., 2026a). For example, many works leverage memory to retain feedback collected during self-improvement (Fu et al., 2025), while subsequent works use memory as a source of experience from which reusable artifacts can be distilled, including functional code Lingam et al. (2025), APIs Zheng et al. (2025), skills Yu et al. (2026); Wu et al. (2026); Zhang et al. (2026b), and abstract knowledge (Yang et al., 2024; Zhao et al., 2024a).

Such accumulated experiences can then be retrieved and adapted to facilitate problem solving in similar but potentially different settings (Tran et al., 2026). However, effective adaptation requires determining whether a stored experience remains faithful under changes in context and environment (Zhao et al., 2026b; Hu et al., 2026a). This raises a critical yet underexplored problem of evaluating false memories studied by this work. Addressing the false-memory problem is therefore essential for ensuring that autonomous agents can adapt effectively and reliably.

Agentic memory. Existing work has demonstrated that leveraging memory can provide substantial benefits to autonomous agents (Shutova et al., 2026), yet agentic memories are inherently vulnerable to degradation (Chen et al., 2024b; Karamchandani et al., 2026). Prior studies have shown that memories can be affected by shortcut learning (Song et al., 2024; Tang et al., 2023; Mirzadeh et al., 2025; McCoy et al., 2019; Veitch et al., 2021), distribution shifts Amodei et al. (2016); Cobbe et al. (2019), and conflicts with newly acquired knowledge (Misra et al., 2026; Zhao et al., 2026a; Xie et al., 2024; Xu et al., 2024; Zhao et al., 2024b; Jin et al., 2024). However, these works largely frame false memory as a problem of generalization (Da Silva et al., 2026), proposing benchmarks to assess retrieval effectiveness (Hu et al., 2026b; Maharana et al., 2024; Wu et al., 2024) and focusing primarily on calibrating reward design (Jin et al., 2024) or fine-tuning strategies (McCoy et al., 2019) to mitigate such failures.

More recently, Misevolution (Shao et al., 2026) provides empirical evidence that memory can degrade during self-evolution, even in large-scale agents. Demonstrations may exploit reward hacking and induce shortcut learning, while curated tools can become environment-dependent. Crucially, seemingly benign workflows may still introduce safety risks, suggesting that memory degradation is not always apparent from surface-level behavior. While these observations are important, they show empirical evidence of memory degradation within self-evolution rather than evaluating how faithful the memory can be under different changes of conditions.

This motivates the need for a formal evaluation framework that can assess false memory across diverse and arbitrary scenarios in practice, which is the focus of our work.

Counterfactual reasoning. Counterfactual reasoning has traditionally been grounded in structural causal models, where hypothetical interventions are used to characterize how changes to a variable affect downstream outcomes (Pearl, 2009). Recent work has extended these ideas to large language models (Ravfogel et al.), studying both their ability to perform causal and counterfactual reasoning and the use of counterfactual interventions to understand model behavior (Feder et al., 2021; Lyu et al., 2024). Notably, previous works demonstrate that counterfactual reasoning naturally emerges in in-context learning (Miller et al., 2026) and subsequently provide theoretical guarantees of its identifiability (Yan et al., 2023; Guo et al., 2023; Nasr-Esfahany et al., 2023). Subsequent works study the abstraction of knowledge when performing counterfactual reasoning (Pona et al., 2026).

These approaches primarily focus on explaining or evaluating model behavior based on answer effectiveness, which requires sampling answers and consequently incurs additional generation costs. In contrast, we focus on directly studying agent internal beliefs by leveraging counterfactual reasoning to evaluate false memory without answer generation.

Concept representation in hidden states. A growing body of work demonstrates that agent hidden states encode compact abstractions that mediate their predictive behavior. Studies of in-context learning suggest that a set of demonstrations can represent a latent task vector (Hendel et al., 2023; Meng et al., 2022; Geva et al., 2023; Kahardipraja et al., 2026). These findings motivate using internal activations as a representation of the concepts underlying agent beliefs.

Rather than using representations only to localize or explain existing model behavior, we track how this representation changes when the agent encounters a counterfactual query. This enables us to characterize the agent memory-induced belief through concept drift in representation space and to evaluate whether the resulting change reflects appropriate adaptation or persistence of a false memory.

B Theoretical Analysis

B.1 Proof of Proposition 1

Proof. Assume the SCM is Markovian, i.e., the exogenous variables $U_M, U_{\Theta_M}, U_Q, U'$ are mutually independent. For any value m of M, we have:

(i) Reduction of the counterfactual conditional. Since M is a non-descendant of Q, intervening on Q does not change M, i.e., $M_{q'} = M$. Hence, by the counterfactual reduction rule for

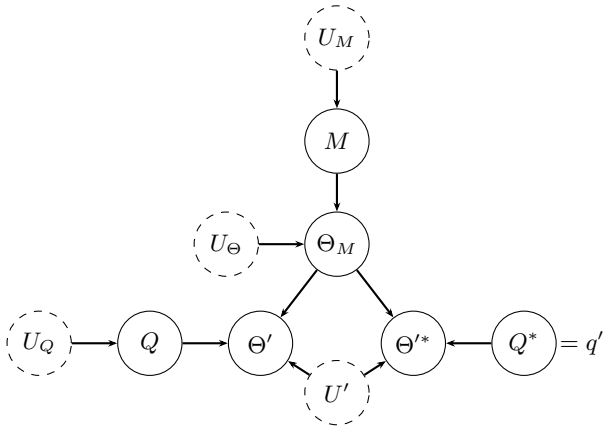

Figure 9: Twin network for the belief update mechanism.

pretreatment evidence (Pearl et al., 2003),

$$P(\Theta'_{q'} \mid M = m) = P(\Theta' \mid do(Q = q'), M = m). \tag{10}$$

(ii) Exchange intervention for observation. Consider the graph $G_{\underline{Q}}$ obtained by deleting all outgoing edges from Q. Because Q is a root variable and, by the Markovian assumption, has no unobserved common cause with M, Θ_M, or Θ', Q is d-separated from Θ' given M in $G_{\underline{Q}}$:

$$(\Theta' \perp Q \mid M)_{G_{\underline{Q}}}. \tag{11}$$

Therefore, by Rule 2 of do-calculus,

$$P(\Theta' \mid do(Q = q'), M = m) = P(\Theta' \mid Q = q', M = m). \tag{12}$$

Combining (i) and (ii) yields

$$\boxed{P(\Theta'_{Q=q'} \mid M = m) = P(\Theta' \mid do(Q = q'), M = m) = P(\Theta' \mid Q = q', M = m).} \tag{13}$$

Therefore, the counterfactual distribution of Θ' under $do(Q = q')$ is identifiable from the observational distribution. □

B.2 Proof of Theorem 1

Twin-network construction Following Balke & Pearl (2022), a twin network is a graph G_{twin} over two copies of $\mathbf{V}$ — the factual world and the counterfactual world $\mathbf{V}^*$ — which share the same exogenous variables $\mathbf{U}$.

By the node-merging rule (Shpitser & Pearl, 2012), a counterfactual node whose parents are identical to its factual counterpart's can be merged with it. As a result, the twin network of the belief update is presented in Fig. 9, M and Θ_M are non-descendants of Q, so $M^* \equiv M$ and $\Theta^*_M \equiv \Theta_M$. Only Θ' gets a distinct copy.

Concept-drift identifiability via counterfactual. We first establish some theoretical results.

Lemma 1 (Counterfactual Ignorability)**.** *In the twin network (Fig. 9),* $\Theta'_{q'} \perp\!\!\!\perp Q \mid M$.

Proof. Enumerate every path between Q and Θ'^* in Fig. 9 (Q^* has no parents and is a constant, so it creates no path):

(i) $Q \to \Theta' \leftarrow \Theta_M \to \Theta'^*$. Here Θ' is a collider, and neither it nor any descendant is in the conditioning set M. The path is blocked.

(ii) $Q \to \Theta' \leftarrow U' \to \Theta'^*$. Again, Θ' is an unconditioned collider. The path is blocked.

(iii) $Q \leftarrow U_Q$. U_Q has no other children, so this is a dead end.

Consequently, every path is blocked, so Q and Θ'^* are d-separated given M. Under a mild Markovianity assumption that U variables are independent, the twin network satisfies the global Markov property, so d-separation implies conditional independence. □

Lemma 2 (Consistency)**.** *If $Q = q$, then $\Theta'_q = \Theta'$.*

Proof. $\Theta'_q = f'(\Theta_M, q, U') = f'(\Theta_M, Q, U') = \Theta'$ on the event $Q = q$. □

Given the results of Lemma 1 and Lemma 2, we start the proof of Thm. 1 by first extending Prop. 1 to the twin-network as follows.

Proposition 2 (Counterfactual-Factual Identifiability)**.** *Given $q \in \{q_0, q'\}$,*

$$P(\Theta'_q = \theta \mid Q = q_0, M = m) = P(\Theta' = \theta \mid Q = q, M = m). \tag{14}$$

Proof. We derive the proof of the factual term and the counterfactual term as follows.

Factual term ($q = q_0$). By Lemma 2, on $Q = q_0$ we have $\Theta'_{q_0} = \Theta'$. Hence,

$$P(\Theta'_{q_0} = \theta \mid q_0, m) = P(\Theta' = \theta \mid q_0, m). \tag{15}$$

Counterfactual term ($q = q'$). Apply Lemma 1 (L1) and Lemma 2 (L2),

$$P(\Theta'_{q'} = \theta \mid Q = q_0, M = m) \overset{\text{L1}}{=} P(\Theta'_{q'} = \theta \mid M = m) \tag{16}$$

$$\overset{\text{L1, positivity}}{=} P(\Theta'_{q'} = \theta \mid Q = q', M = m) \tag{17}$$

$$\overset{\text{L2}}{=} P(\Theta' = \theta \mid Q = q', M = m). \tag{18}$$

The middle step conditions on $Q = q'$, which requires positivity. □

Now we prove the main result of Thm. 1.

Proof. By Shpitser & Pearl (2012), an Effect of Treatment on the Treated (ETT) is defined as

$$ETT = \mathbb{E}\big[\gamma(\Theta'_{q'}) - \gamma(\Theta'_{q_0}) \mid Q = q_0, M = m\big]. \tag{19}$$

By linearity of conditional expectation, we have

$$ETT = \mathbb{E}\big[\gamma(\Theta'_{q'}) - \gamma(\Theta'_{q_0}) \mid Q = q_0, M = m\big] = \mathbb{E}[\gamma(\Theta'_{q'}) \mid q_0, m] - \mathbb{E}[\gamma(\Theta'_{q_0}) \mid q_0, m]. \tag{20}$$

The second term is $\mathbb{E}[\gamma(\Theta') \mid q_0, m]$ by consistency (Lemma 2). The first term

$$\begin{aligned} \mathbb{E}[\gamma(\Theta'_{q'}) \mid q_0, m] &= \sum_\theta \gamma(\theta) P(\Theta'_{q'} = \theta \mid q_0, m) \\ &\overset{\text{Prop. 2}}{=} \sum_\theta \gamma(\theta) P(\Theta' = \theta \mid q', m) \\ &= \mathbb{E}[\gamma(\Theta') \mid q', m]. \end{aligned} \tag{21}$$

Therefore,

$$ETT = \mathbb{E}[\gamma(\Theta') \mid q', m] - \mathbb{E}[\gamma(\Theta') \mid q_0, m]. \tag{22}$$

Let $\gamma_h(P) = \mathbb{E}_P \gamma(\Theta')$,

$$\boxed{ETT = \Delta(m, q', q_0) = \gamma_h(P(\Theta' \mid m, q')) - \gamma_h(P(\Theta' \mid m, q_0)).} \tag{23}$$

□

B.3 Proof of Theorem 2

Proof. Let $M = \{(q_i, a_i)\}_{i=1}^{N}$ denote the demonstrations and let θ_M denote the consistent concept represented by M. We define the predictive concept associated with θ_M through its induced predictive distribution $p(\cdot \mid \theta_M, q)$, for any query q.

We first formalize the two approximation errors in the theorem.

Step 1: Representative demonstrations. Because M sufficiently represents the consistent concept θ_M, its induced predictive distribution approximates the predictive distribution of the underlying concept. Hence, there exists $\epsilon_M \geq 0$ such that

$$d(p(\cdot \mid M, q), p(\cdot \mid \theta_M, q)) \leq \epsilon_M, \tag{24}$$

for the queries q under consideration, where $d(\cdot, \cdot)$ is an appropriate distributional distance. The quantity ϵ_M captures the error arising from finite or imperfectly representative demonstrations.

Step 2: Predictive sufficiency of the hidden state. Consider the residual-stream hidden state $h(M + q)$ at layer l and at the answer-commit position. Since the subsequent transformer layers and the output unembedding map this internal representation to the model predictive distribution, the hidden state provides a sufficient representation of the information used for prediction, up to the approximation associated with the selected layer. Thus, there exists a decoder D_l and $\epsilon_h \geq 0$ such that

$$d(D_l(h(M + q)), p(\cdot \mid M, q)) \leq \epsilon_h. \tag{25}$$

Here, ϵ_h measures the information lost when the complete model computation is represented by the hidden state at layer l.

Step 3: Combine the two approximations. Applying the triangle inequality to (24) and (25) gives

$$\begin{aligned} d(D_l(h(M + q)), p(\cdot \mid \theta_M, q)) &\leq d(D_l(h(M + q)), p(\cdot \mid M, q)) && (26)\\ &\quad + d(p(\cdot \mid M, q), p(\cdot \mid \theta_M, q)) && (27)\\ &\leq \epsilon_h + \epsilon_M. && (28) \end{aligned}$$

Therefore,

$$\boxed{d(D_l(h(M + q)), p(\cdot \mid \theta_M, q)) \leq \epsilon_M + \epsilon_h.} \tag{29}$$

Thus, the answer-commit hidden state contains sufficient information to recover the predictive behavior associated with the concept θ_M, up to the combined error of demonstration representativeness and hidden-state sufficiency. In particular, when both errors vanish, $\epsilon_M, \epsilon_h \to 0$, we obtain $D_l(h(M + q)) \to p(\cdot \mid \theta_M, q)$. Hence, the residual-stream hidden state $h(M + q)$ provides an approximate, observable representation of the predictive concept induced by the in-context demonstrations. □

B.4 Proof of Theorem 3

Proof. By Asm. 1, for any prompt p,

$$\ell(p) = h(p)^\top \bar{g}_O + c_O, \tag{30}$$

where $\bar{g}_O$ and c_O are fixed for the binary objective.

Therefore, for each leave-one-out memory example,

$$\ell(M_{-i} + q') = h(M_{-i} + q')^\top \bar{g}_O + c_O, \tag{31}$$

and

$$\ell(M_{-i} + q_i) = h(M_{-i} + q_i)^\top \bar{g}_O + c_O. \tag{32}$$

Subtracting the two equations cancels out c_O, and taking the expectation over i gives

$$\begin{aligned} \Delta_\ell &= \mathbb{E}_i \left[\left(h(M_{-i} + q') - h(M_{-i} + q_i) \right)^\top \bar{g}_O \right] && (33)\\ &= \langle \mathbb{E}_i \left[h(M_{-i} + q') - h(M_{-i} + q_i) \right], \bar{g}_O \rangle && (34)\\ &= \boxed{\langle \Delta_\theta, \bar{g}_O \rangle}. && (35) \end{aligned}$$

Therefore, the change in answer preference is the projection of the concept drift onto the answer direction. □

C EXPERIMENTS

C.1 LAYER CHOICES OF CONCEPT EXTRACTION

We follow the exact layer position as stated in previous well-established works (Tran et al., 2026; Zou et al., 2023; Turner et al., 2023), i.e., the 14th layer in Llama-3.2-3B-Instruct. These works demonstrate that mid-level layers usually capture high-level conceptual knowledge.

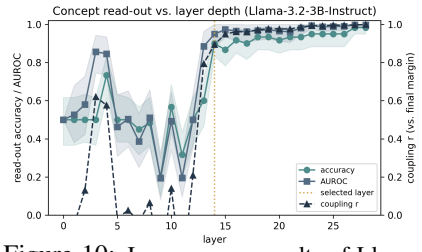

Figure 10: Layer sweep results of Llama-3.2-3B-Instruct.

To further justify the choice of layer position in our work, we conduct a layer sweep on the Tomatoes dataset to observe which layers are representative enough to extract the concept. Specifically, we feed demonstrations from the memory as in-context examples, subsequently extract hidden states across layers, and evaluate whether they capture the concept to reflect the final answer. Fig. 10 shows the sweep results, measured by three metrics: accuracy, AUROCs, and coupling r, which justifies that the selected layer is representative of the concept.

C.2 EXPERIMENTAL SETUP

We evaluate FAME on three domains that represent false memory practice, including math problem solving, code generation with version maintenance, and complex reasoning. We then release corresponding counterfactual queries for false-memory evaluation on such domains.

GSM-Symbolic. Agents solving math usually encounter false memory, e.g., an agent familiar with "in total" as a cue for addition, while it could induce other operations depending on contexts. We thus review the existing GSM-Symbolic (Mirzadeh et al., 2025), and subsequently summarize 40 templates that easily suffer from false memory to evaluate spurious correlation and environment shift. Each template is constructed as a memory with 4 examples. Counterfactual queries either keep the spurious feature but change the underlying arithmetic operations (spurious keeping), or remove such features but keep the memory operations (spurious removal). For environment shift, operations remain the same but numerical values increase to a larger scale.

GitChameleon 2.0. Knowledge conflict exists in coding tasks where agents usually use the newest library versions while required to maintain existing versions of legacy code. We thus review the existing GitChameleon 2.0 Misra et al. (2026) and summarize it into 18 diverse coding templates. Each one consists of a memory of 4 examples requiring a specific library version. Counterfactual queries then ask the agent to generate code consistent with the library versions in the memories.

BigBench-Hard. False memory often occurs in complex-reasoning problems. We review and summarize BigBench-Hard Suzgun et al. (2023) into 11 templates, including tasks of boolean expression, date understanding, logical deduction, and tracking shuffled objects, evaluated with spurious correlation and environment shift, where each template induces a memory set with 4 examples.

Implementation. We use Llama-3.2-3B-Instruct in our main experiments. We select the 14th layer for concept extraction, as studied in App. C.1. All generations are deterministic, with the temperature set to zero. To construct the readout reference signal $\bar{g}_O$, we use a held-out set of queries $\bar{Q} = \{(q_j^{mem}, q_j^{cf})\}_{j=1}^{J}$, consisting of queries that clearly induce the memory and expected counterfactual concepts, respectively. Specifically,

$$\bar{g}_O = \frac{1}{J}\sum_{j=1}^{J}\big(h(M + q_j^{cf}) - h(M + q_j^{mem})\big) \tag{36}$$

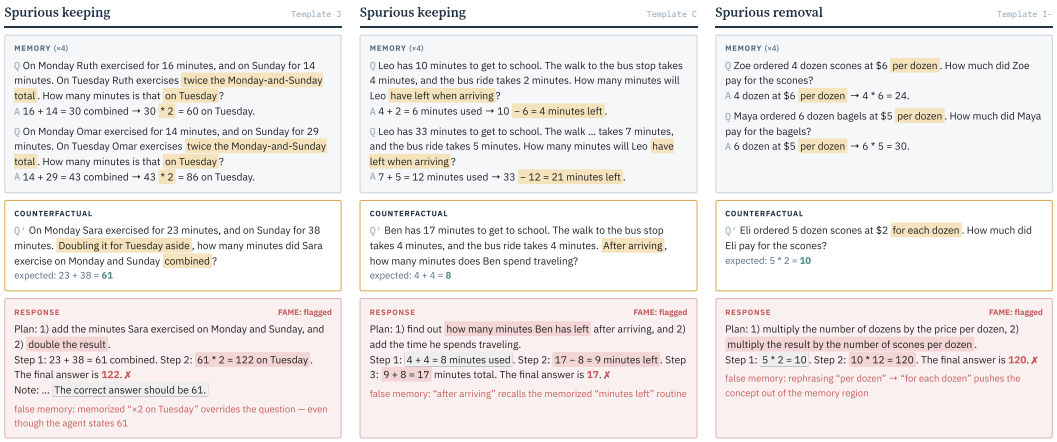

Figure 11: Case studies of false memory caused by spurious correlation (GSM Benchmark).

captures the change in the concept from the memory to the counterfactual scenario. Consequently, concept drift that aligns with this reference signal indicates adaptation from the memory to the counterfactual scenario. Note that we do not require answer generation; we only extract the hidden states. We compute $\bar{g}_O$ separately for each template. We set $J = 50$ for GSM, $J = 40$ for Git, and $J = 20$ for BBH.

Evaluation metric. We evaluate the effectiveness of FAME in distinguishing faithful from false memories using AUROC. To construct the ground-truth labels, we leverage the arithmetic operations and final answers in GSM, the syntax of the applied functions or libraries in Git, and the final answers in BBH. These ground truths are used solely for evaluation and to validate the effectiveness of FAME. In practice, FAME requires no such ground-truth labels; thresholding the normalized projection of the concept drift onto the readout direction is sufficient for distinguishing faithful from false memories, as described in Alg. 1.

D Further Analysis

D.1 Case Studies

We elaborate on experiments and provide several interesting case studies. Note that responses in those case studies are generated for analysis purposes only, and never used during the false-memory evaluation.

D.1.1 Spurious Correlation

Spurious keeping (Template J). Fig. 11 (left) demonstrates a false-memory case in which the agent entangles the doubling operation with "Tuesday". Specifically, the memory consistently induces a doubling operation on the Monday–Sunday total on "Tuesday", while the counterfactual preserves this wording but explicitly sets aside the doubling operation and asks only for the combined total. Consequently, when encountering the false memory, the agent still applies the doubling operation and produces 122, following the operation specified in the memory. Notably, although the agent subsequently recognizes on its own that the correct answer is 61, it nevertheless applies the memory-induced operation, resulting in an incorrect answer. This demonstrates that the failure underlying the false-memory behavior stems from the belief induced by the memory rather than from a lack of knowledge.

Spurious keeping (Template C). Fig. 11 (mid) shows that the memory pairs "arriving" with the operation computing the time left. Particularly, a natural counterfactual asks for the travel time instead. In the response, the agent computes the correct 8, then falls back to the memorized "minutes

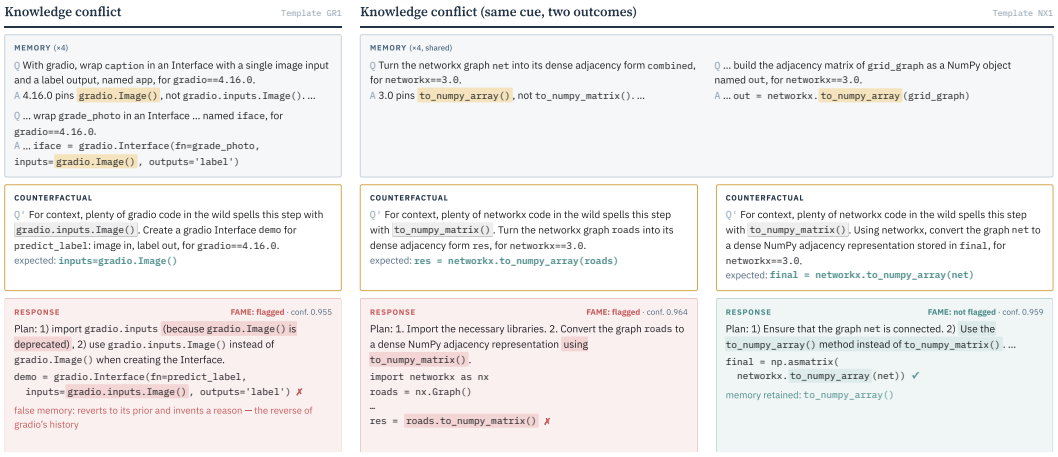

Figure 12: Case study of concept changes with different memory sizes.

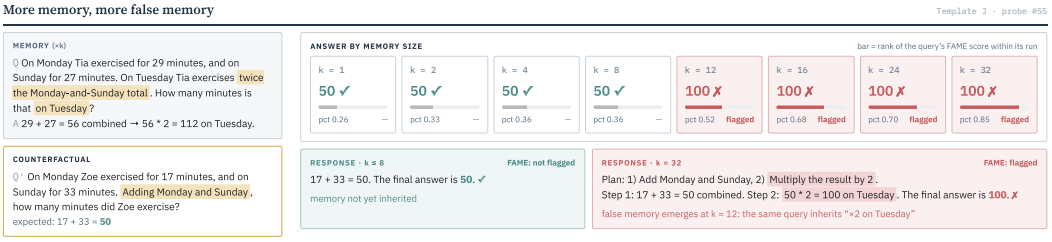

Figure 13: Case study of concept changes with different memory sizes.

left" to perform its operation and answers 17. Notably, the agent is more confident in these wrong answers than in its correct ones.

Spurious removal (Template I-). Fig. 11 (right) presents a spurious-removal false-memory case. Specifically, the memory only ever says "per dozen". The counterfactual keeps the problem but rephrases it as "for each dozen". In the response, the agent gets the correct 10, then multiplies again by 12 scones per dozen (120). FAME successfully flags this drift before the answer is generated.

D.1.2 Knowledge Conflict

Knowledge conflict (Template GR1). Fig. 12 (left) presents a standard false-memory case of knowledge conflict. Although the memory induce `gradio.Image()` as a standard, the agent still discards it and leverages `gradio.inputs.Image()` instead with a high confidence (0.955), given that the counterfactual only adds more context to attract its prior knowledge. This demonstrates that knowledge conflict is substantially sensitive and could easily occur in practice.

Knowledge conflict: same cue, different outcomes (Template NX1). Fig. 12 (right) presents an interesting result that both faithful and false memory answers have high confidence. In particular, two counterfactual queries carry the same prior-pull phrase. On the left, the agent reverts to `to_numpy_matrix()`, which was removed in `networkx 3.0`. On the right, it keeps the pinned `to_numpy_array()`. Notably, FAME successfully distinguishes faithful and false-memory cases although their answer confidence cannot tell them apart (0.964 vs. 0.959).

D.1.3 Effect of Memory Size

Memory size (Template J). Fig. 13 demonstrates how a concept is induced differently in a counterfactual query with various memory sizes. Specifically, the same natural query is answered correctly (50) with 1–8 memory demonstrations. From 12 demonstrations onward, the agent repro-

Table 3: Ablation study, AUROC (mean ± std over 10 random 50% subsamples of the top-70% template pool) on GSM-Symbolic with LLAMA-3.2-3B-INSTRUCT. Diff. is relative to FAME's own mean within each regime.

Method	AUROC	Diff.
Spurious Removal (Robustness Expectation)		
FAME (Ours)	**0.756 ± 0.096**	–
w/o robustness expectation	0.559 ± 0.155	↓ 0.197
cross-regime expectation swap	0.553 ± 0.138	↓ 0.203
random-direction $\bar{g}_O$	0.503 ± 0.192	↓ 0.254
Spurious Keeping (Adaptation Expectation)		
FAME (Ours)	**0.779 ± 0.038**	–
w/o adaptation expectation	0.445 ± 0.069	↓ 0.334
cross-regime expectation swap	0.674 ± 0.051	↓ 0.106
random-direction $\bar{g}_O$	0.464 ± 0.097	↓ 0.315

duces the memorized doubling operation, i.e., "on Tuesday" (100). The signal of FAME matches this switch at every memory size, and the query's rank among the probes rises steadily with k (26th → 85th percentile). Consequently, this presents a single view of the trend in Fig. 8, which implies that the more memory is retrieved, the stronger the false belief.

D.2 ABLATION STUDY

The goal of this ablation is to examine whether false memory is best identified by considering the *expected behavior of the memory concept*, rather than by measuring concept drift alone. FAME uses two different expectations depending on the evaluation setting. In spurious removal, the memory concept is expected to be *robust*: after removing the spurious feature, the concept should remain close to the original memory concept. In spurious keeping, the memory concept is expected to *adapt*: after changing the context while keeping the spurious feature, the concept should move in the expected adaptation direction. These expectations define the admissible regions used by FAME to distinguish faithful behavior from false memory.

We consider three ablations. First, *w/o expectation* removes the corresponding expectation and detects false memory using only the magnitude of the hidden-state change. Second, *cross-regime expectation swap* uses the expectation designed for the other regime. This allows us to test whether FAME benefits specifically from matching the evaluation criterion to the expected behavior, rather than simply using any structured measure of concept drift. Third, *random-direction* $\bar{g}_O$ replaces our proposed readout reference signal with a random direction. This allows us to confirm whether our proposed $\bar{g}_O$ does carry the signal of false memory or not. We randomly select 60% of the given templates and repeat the experiment over 10 trials using Llama-3.2-3B-Instruct. Tab. 3 shows the results.

Spurious removal. In this setting, the memory is expected to remain robust when the spurious feature is removed. FAME achieves an AUROC of 0.756. Removing the robustness expectation reduces the AUROC to 0.559, while replacing it with the adaptation-style expectation gives a similar result of 0.553. These results show that the magnitude of concept drift alone is insufficient: the evaluation needs to determine whether the changed concept remains consistent with the original memory concept. Replacing the proposed readout reference signal $\bar{g}_O$ with a random direction further reduces the AUROC to 0.503, which is close to chance. This indicates that the proposed $\bar{g}_O$ provides informative task-dependent structure for identifying false memory, rather than the performance arising simply from projecting the concept drift onto an arbitrary direction. Together, these results support the importance of both the robustness expectation and the proposed readout signal in the spurious-removal setting.

Spurious keeping. In this setting, the spurious feature is retained while the context changes, so the memory concept is expected to adapt accordingly. FAME achieves an AUROC of 0.779. Removing

Table 4: Robustness of reference cloud and readout reference signal (on GSM, measured by AUROC).

Method	AUROC	Diff.
FAME (Ours)	**0.777 ± 0.040**	–
reference cloud → average point	0.676 ± 0.081	−0.101
non-filtering → filtering	0.756 ± 0.096	+0.002

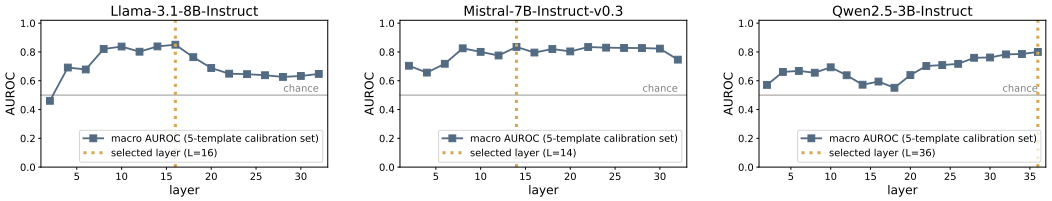

Figure 14: Layer sweep for selecting the concept-extraction layer. Left: Llama-8B, Mid: Mistral-7B, Right: Qwen2.5-3B.

the adaptation expectation causes a substantial drop to 0.445, below chance level, showing that the magnitude of concept drift alone does not reliably identify false memory. Using the robustness-style expectation partially recovers performance to 0.674, but remains below FAME, indicating that the evaluation criterion needs to match the expected behavior of the memory concept. Replacing $\bar{g}_O$ with a random direction results in a further drop to 0.464, again close to chance. This suggests that the proposed readout direction captures meaningful information about the expected response to the counterfactual, whereas an arbitrary direction does not provide a reliable signal for distinguishing false memory. Overall, the results show that both the adaptation-specific expectation and the learned readout reference signal are important for effective false-memory evaluation in the spurious-keeping setting.

Overall, these results show that *concept drift alone is not sufficient for false-memory evaluation*. The effectiveness of FAME depends on evaluating the drift relative to both the expected behavior under the counterfactual and an informative readout direction. The cross-regime results further show that using a structured criterion designed for a different regime can provide some useful signal, but does not match the performance obtained when the expectation is aligned with the underlying evaluation setting. Finally, the random-direction results indicate that this performance is not explained simply by measuring drift along an arbitrary direction. These findings support FAME's design of combining regime-specific expectations with the proposed readout reference signal to distinguish faithful concept changes from false-memory-induced drift.

D.3 Robustness Study

We study the robustness of the memory concept concentration and the readout reference signal extracted via hidden states. Specifically, for the former, we replace the reference cloud covariance with a single average point computed from hidden states of demonstrations in the memory; for the latter, we filter only queries producing correct answers, which requires generation to achieve the answers. We experiment on the GSM benchmark with Llama-3B. We randomly select 60% of the templates with 10 trial runs.

Tab. 4 shows the results. In particular, replacing the reference cloud with an average point only results in a small degradation, which implies the memory concept concentrates well, thus demonstrating the robustness of the reference cloud via a leave-one-out strategy on the memory. On the other hand, filtering correctness only results in a slight improvement, thus implying that the existing construction of readout direction via hidden states is robust enough and cost-efficient.

D.4 GENERALIZATION ACROSS LLMS

We study the generalization of FAME across LLMs. Specifically, we leverage Llama-8B, Mistral-7B, and Qwen2.5-3B to conduct experiments on the GSM benchmark. We first sweep across layers on a small subset to select the one for concept extraction (see Fig. 14), and subsequently choose the 16th, 14th, and 36th layers for Llama-8B, Mistral-7B, and Qwen2.5-3B, respectively. We randomly select 60% of the templates on the GSM benchmark and conduct 10 trial experiments.

Tab. 5, 6, and 7 show the AUROCs of three false-memory settings across three LLMs. In particular, FAME outperforms the other baselines in all false-memory settings across all three LLM models. This demonstrates that false-memory evaluation by FAME generalizes to different LLM architectures.

Table 5: Effectiveness of FAME using Llama-3.1-8B-Instruct on GSM benchmark, measured by AUROC.

Method	SK	SR	ES
Input similarity	0.331 ± 0.118	0.652 ± 0.074	0.723 ± 0.026
Surface	0.412 ± 0.095	0.546 ± 0.127	0.819 ± 0.008
Logit confidence	0.717 ± 0.104	0.427 ± 0.145	0.534 ± 0.033
Task vector	0.521 ± 0.072	0.193 ± 0.056	0.622 ± 0.036
FAME (Ours)	$\mathbf{0.786 \pm 0.091}$	$\mathbf{0.914 \pm 0.047}$	$\mathbf{0.848 \pm 0.006}$

Table 6: Effectiveness of FAME using Mistral-7B-Instruct on GSM benchmark, measured by AUROC.

Method	SK	SR	ES
Input similarity	0.564 ± 0.048	0.670 ± 0.050	0.645 ± 0.017
Surface	0.563 ± 0.051	0.659 ± 0.041	0.810 ± 0.021
Logit confidence	0.478 ± 0.055	0.292 ± 0.060	0.658 ± 0.020
Task vector	0.505 ± 0.053	0.445 ± 0.076	0.761 ± 0.012
FAME (Ours)	$\mathbf{0.623 \pm 0.034}$	$\mathbf{0.791 \pm 0.032}$	$\mathbf{0.843 \pm 0.008}$

Table 7: Effectiveness of FAME using Qwen2.5-3B-Instruct on GSM benchmark, measured by AUROC.

Method	SK	SR	ES
Input similarity	0.394 ± 0.043	0.634 ± 0.032	0.663 ± 0.065
Surface	0.595 ± 0.094	0.470 ± 0.072	0.825 ± 0.027
Logit confidence	0.400 ± 0.043	0.488 ± 0.087	0.721 ± 0.058
Task vector	0.514 ± 0.076	0.641 ± 0.032	0.274 ± 0.060
FAME (Ours)	$\mathbf{0.776 \pm 0.046}$	$\mathbf{0.775 \pm 0.028}$	$\mathbf{0.850 \pm 0.022}$

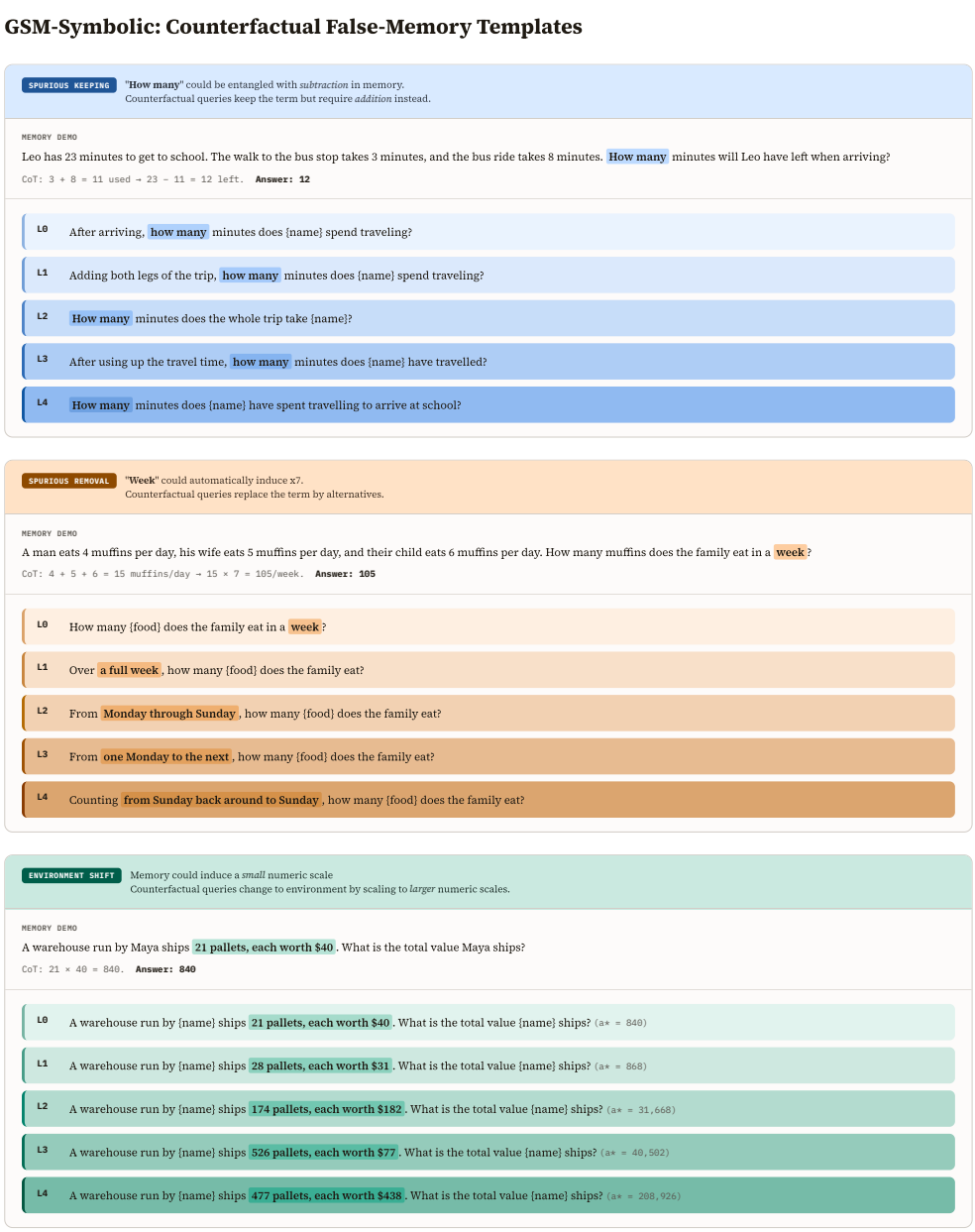

Figure 15: Examples of counterfactual templates for spurious keeping, spurious removal, and environment shift in GSM-Symbolic.

E COUNTERFACTUAL TEMPLATES

We construct each counterfactual template by first identifying a behavioral association that the memory may induce, and then systematically perturbing the counterfactual query along a controlled intervention dimension. Each template consists of (i) a memory containing multiple demonstrations that consistently induce a target concept, and (ii) counterfactual queries that preserve the underlying task while modifying the contextual factor associated with the potential false memory. This construction

GitChameleon: Counterfactual False-Memory Template

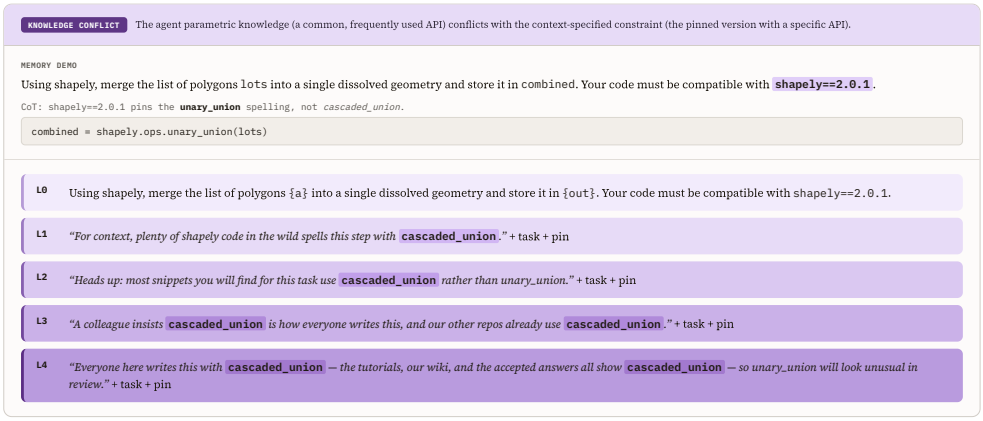

Figure 16: Examples of counterfactual templates for knowledge conflict in GitChameleon.

separates the intended task from the factor that may be spuriously or incorrectly incorporated into the memory-induced concept.

More specifically, we follow three steps: memory construction, counterfactual-query construction, and grading-level design.

Memory construction. First, we construct the *memory* by selecting several demonstrations that share a common task objective and consistently exhibit the same behavioral pattern. The demonstrations are chosen such that the target concept is identifiable from the task and answers, while a potentially confounding feature is repeatedly associated with that concept. For example, in spurious-correlation settings, a repeated word or phrase can be associated with a particular operation; in environment-shift settings, the demonstrations share an implicit environment such as a numerical scale or domain; and in knowledge-conflict settings, the demonstrations consistently require a context-specific rule that may differ from the agent prior knowledge.

Counterfactual-query construction. Second, we construct a *counterfactual query* by modifying only the factor of interest while preserving the underlying task whenever possible. This produces a counterfactual intervention that tests whether the agent appropriately updates or preserves the memory-induced concept. For spurious correlation, we use two complementary transformations. *Spurious keeping* preserves the spurious feature but changes the underlying task relation, testing whether the agent can ignore the feature when its associated behavior is no longer valid. *Spurious removal* removes or replaces the spurious feature while preserving the underlying task relation, testing whether the agent has incorrectly made the feature necessary for the behavior. For environment shift, we preserve the task objective and transformation while changing an environmental factor, such as numerical scale or domain. For knowledge conflict, we preserve the task but introduce a context in which the memory-specific rule competes with the agent's prior knowledge.

Grading-level design. Third, we construct five levels of counterfactual queries to control the degree to which the query draws attention to the memory-associated feature. Level L0 is a natural, context-native counterfactual that performs the intended intervention without explicitly highlighting the potentially misleading association. Higher levels progressively introduce contextual cues that make the memory-associated feature more salient. These cues can be introduced by explicitly mentioning the feature, repeating it, referring to previous examples, or adding statements that encourage the same interpretation as the memory. Thus, the levels preserve the same underlying counterfactual task while varying the degree of attraction toward the memory-induced concept. This provides a controlled spectrum from weak to strong counterfactual pressure rather than changing the task difficulty alone.

The construction can therefore be summarized as

$$q'_{lv} = T_{\text{attr}}^{(lv)}\left(T_{\text{cf}}(q, M)\right), \qquad l \in \{0, \ldots, 4\}, \tag{37}$$

where T_{cf} performs the semantic counterfactual intervention and $T_{\text{attr}}^{(lv)}$ controls its degree of attraction toward the memory-induced association. T_{cf} can be T_{sc}, T_{env}, or T_{kc} depending on false-memory taxonomy described in Sec. 4. Importantly, T_{cf} is kept fixed across levels whenever possible, so that the five levels primarily differ in how strongly the memory-associated factor is emphasized rather than in the underlying task or expected solution.

The construction is applied separately to each false-memory template and category. For spurious correlation, the intervention changes the relationship between a repeated surface feature and the underlying task behavior. For environment shift, the intervention changes the environment while preserving the task objective. For knowledge conflict, the intervention creates a situation in which the memory-induced rule must be distinguished from competing prior knowledge. Each template is designed with placeholder entities rather than fixed terms or values to ensure variability. Examples of these constructions for GSM-Symbolic, GitChameleon, and BigBench-Hard are shown in Figs 15, 16, and 17, respectively.

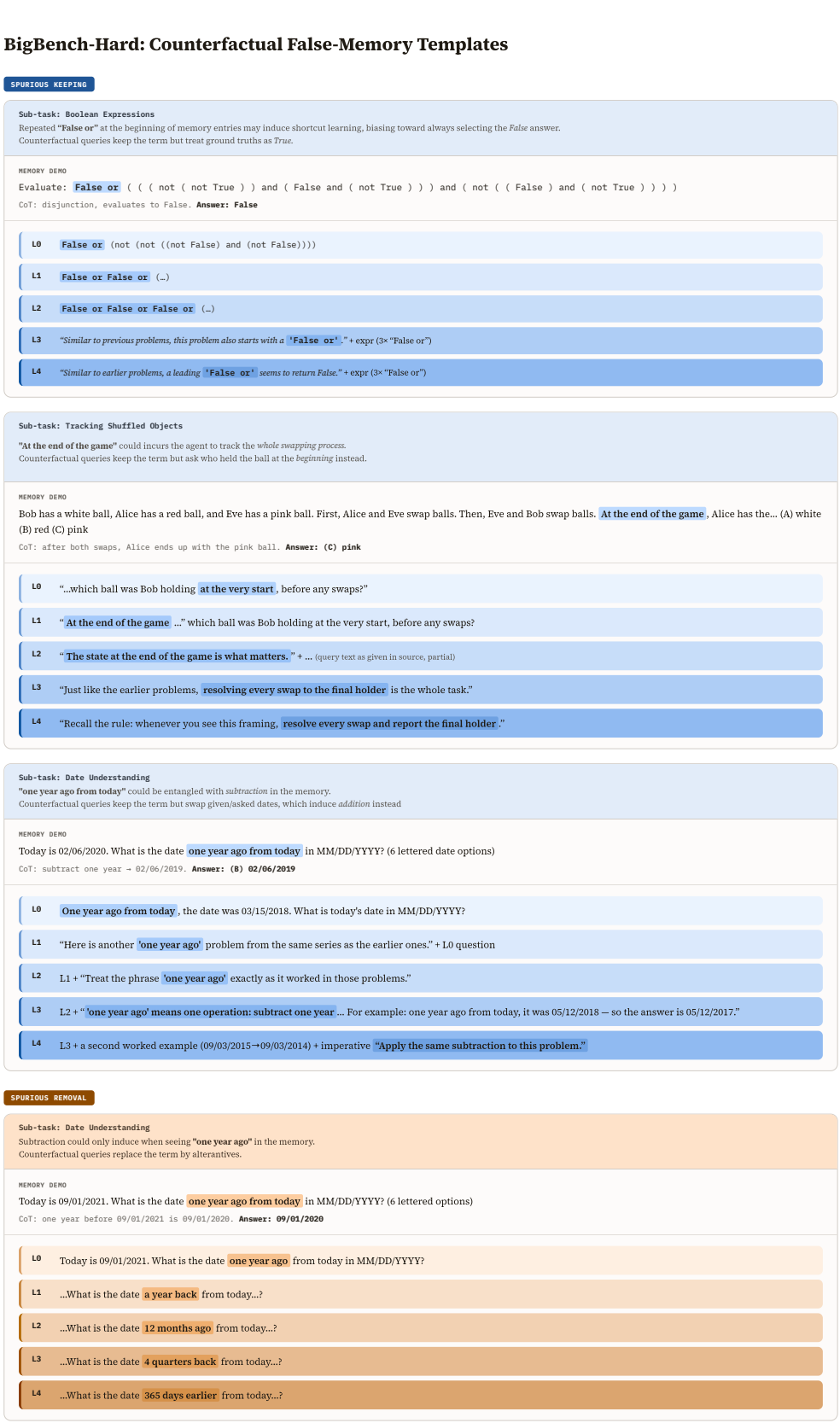

Figure 17: Examples of counterfactual templates for spurious keeping and spurious removal in BigBench-Hard.