SAVER: SEQUENTIAL ANYTIME-VALID EDITING RISK-CONTROL IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential knowledge editing (KE) in large language models (LLMs) can accumulate interference, weakening both edit success and preservation of reference behavior. Existing editing methods improve update quality and reduce interference, but failures in individual updates can still go undetected. Once committed, these updates alter the starting point of subsequent edits, allowing their effects to persist and accumulate. To address this, we propose SAVER (Sequential Anytime-Valid Editing Risk-control), a model- and editor-agnostic pre-commit controller for sequential KE. Specifically, SAVER measures prediction-set miscoverage on generality and locality probes and uses sequential risk evidence to select the smallest admissible boundary and guide edit admission. We further develop interference-aware sampling with a control-variate risk estimator to reduce full-probe evaluation cost. Moreover, we establish anytime-valid false-alarm control across time and candidate boundaries, together with detection-delay bounds under persistent excess risk and the stated monitoring conditions. Extensive experiments across editors, backbone LLMs, and datasets demonstrate improved reference preservation while retaining useful updates and reducing monitoring cost. Our code is available at `https://anonymous.4open.science/r/SAVER-E7D6/`.

1 INTRODUCTION

Knowledge Editing (KE) for Large Language Models (LLMs) (Meng et al., 2022; 2023; Mitchell et al., 2022) aims to update factual information encoded in pre-trained models without full retraining. In sequential editing, these updates arrive as a stream, requiring the model to incorporate new facts while preserving behavior on knowledge that should remain unchanged. However, successive edits modify shared model parameters, allowing earlier updates to interfere with later edits and alter behavior that should remain stable. As illustrated in Figure 1, repeated edits can weaken previously updated knowledge and progressively degrade model behavior (Li & Chu, 2024; Yang et al., 2024b).

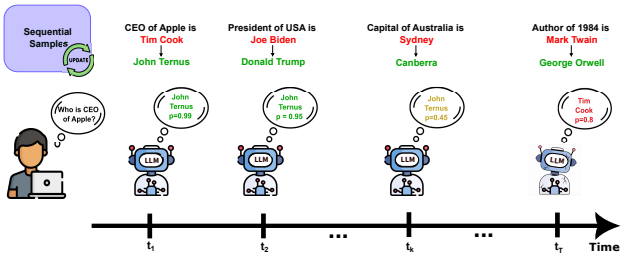


Figure 1: Sequential editing with updates arriving as a continuous edit stream. Over time ($t_1 \rightarrow t_T$), repeated accepted edits may perturb related facts, weaken previous edits, and degrade the model's initial reference behavior.

In recent years, researchers have put much effort in improving editing performance in continual settings (Wang et al., 2024; Fang et al., 2024; Li & Chu, 2025; Gu et al., 2026a). These methods primarily improve how updates are constructed, aiming to incorporate successive factual changes while preserving existing knowledge. However, uncertainty-based diagnostics show that an edit can appear successful under conventional success and locality measures while introducing hidden instability (Gu et al., 2026b). Studies of editing-induced degradation further show that even a single edit can substantially impair the broader model capabilities (Yang et al., 2024b). In sequential editing, these effects extend beyond the current update as each accepted edit changes the model state on which subsequent edits operate. Failures that remain undetected can therefore persist and interact with later updates, undermining both the incorporation of new knowledge and preservation of reference behavior. This motivates us to ask:

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.

Can we quantify and monitor knowledge editing risk with statistical guarantees to retain useful knowledge updates while limiting degradation across successive edits?

To answer this question, we propose Sequential Anytime-Valid Editing Risk-control (SAVER), a model- and editor-agnostic pre-commit controller that jointly assesses edit success and preservation of reference behavior to decide which candidate updates enter the committed edit stream. However, developing such a controller raises certain challenges. First, an edit specifies a target fact but does not provide the desired response for every input affected by the update. Changes beyond edit prompt may reflect successful generalization or unintended interference, making it difficult to determine which behavioral changes should count as editing failures. Second, understanding these effects requires evaluating the model beyond edited fact, making a thorough assessment increasingly costly over long edit streams. Most importantly, it remains to be seen how evidence about these effects can guide which updates to commit with statistical guarantees, when each accepted update changes the model and conditions faced by subsequent edits.

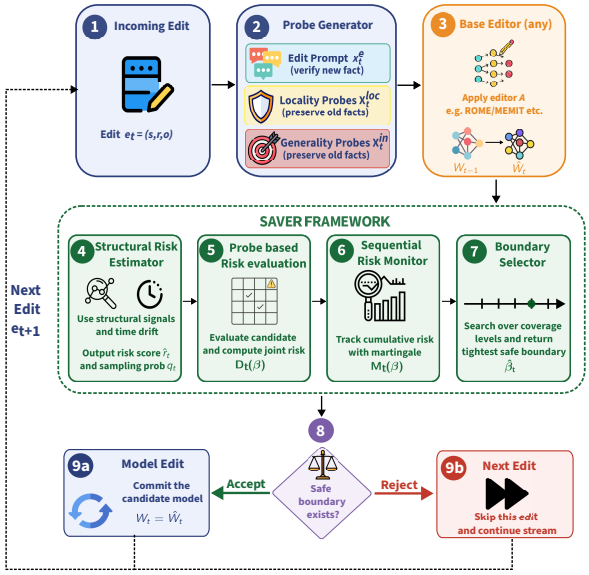


Figure 2: Overview of the proposed SAVER framework. Interference-aware sampling prioritizes candidates with higher predicted risk for full probe evaluation. Sequential monitors use corrected risk estimates to select the smallest admissible boundary, committing the candidate if one exists and rejecting it otherwise.

SAVER addresses these challenges by pairing generality probes with the edited target and locality probes with the initial model's reference responses. It then measures the exclusion of these targets from the candidate's prediction sets, resulting in a joint risk with an explicit evaluation scope and failure criterion. Next, a lightweight proxy combines semantic similarity to previously committed edits with the time since the last full evaluation to prioritize candidates for probe evaluation. A control-variate importance estimator combines this proxy with sampled probe losses to obtain conditionally unbiased risk estimates. Finally, these estimates drive a sequential martingale monitor that tracks evidence of excess risk for each candidate prediction-set boundary. SAVER selects the smallest boundary whose monitor is below the simultaneous rejection threshold and commits the candidate update when such a boundary exists; otherwise, it retains the last committed model. We establish anytime-valid false-alarm control across time and the boundary grid, and characterize the detection delay under persistent excess risk and the stated monitoring conditions. The overall framework is depicted in Figure 2. Our main contributions are summarized as follows:

- We formulate sequential knowledge editing as an online risk-control problem using a probe-defined joint risk over edit success and preservation of initial-model reference behavior. Accordingly, we propose SAVER, a model- and editor-agnostic pre-commit controller that wraps existing editors and uses sequential risk evidence to guide admission.
- To implement SAVER efficiently across long edit streams, we develop interference-aware sampling that prioritizes candidates with higher predicted risk for full-probe evaluation. We also establish conditional unbiasedness and variance-control guarantees for the resulting sampled risk estimator (Lemma 1 and theorem 1).
- Theoretically, we establish anytime-valid false-alarm control across time and candidate boundaries (Theorem 3 and corollary 1), and bound the expected detection delay when the probe-defined joint editing risk persistently exceeds the user-specified threshold (Theorem 4).

- Finally, we conduct comprehensive experiments across editors, backbone LLMs, datasets, and edit-stream lengths (Section 4), demonstrating the effectiveness and efficiency of SAVER in limiting sequential degradation while retaining useful updates.

2 The Proposed Framework

In this section, we formulate **S**equential **A**nytime-**V**alid **E**diting **R**isk-control (SAVER) as a pre-commit controller for sequential parameter editing. We consider a large language model (LLM) defined over a parameter space $\mathcal{W}$, where $W_0 \in \mathcal{W}$ denotes the initial pre-trained weights. The model receives a lifelong stream of knowledge edits $\mathcal{E} = \{e_t\}_{t=1}^{T}$, where each edit $e_t = (s, r, o)$ represents a factual triple consisting of a subject s, a relation r, and a target object o. To assess edit success and preservation of reference behavior, we define a probe generator $\mathcal{I}$ that, for each edit e_t, produces an edit prompt x_t^e and two disjoint sets of probes: locality probes $\mathcal{X}_t^{loc}$ and generality probes $\mathcal{X}_t^{in}$ [1]. Specifically:

$$x_t^e, \mathcal{X}_t^{in}, \mathcal{X}_t^{loc} = \mathcal{I}(s, r, o). \tag{1}$$

Generality probes $\mathcal{X}_t^{in}$ assess whether the requested factual change generalizes beyond the edit prompt whereas the locality probes $\mathcal{X}_t^{loc}$ quantify preservation of the initial model's behavior. To evaluate a candidate edit, each probe x is paired with an evaluation target $y_t^*(x)$ such that:

$$y_t^*(x) = \begin{cases} o & \text{if } x \in \mathcal{X}_t^{in}, \\ y_{base}(x) & \text{if } x \in \mathcal{X}_t^{loc}. \end{cases} \tag{2}$$

Here o is the target object of edit e_t, and $y_{base}(x)$ is the reference token predicted by the initial model W_0 on probe x. Given the edit prompt x_t^e, a base editing algorithm $\mathcal{A}$ (e.g., ROME or MEMIT) produces candidate weights $\hat{W}_t$ from the last committed weights W_{t-1}:

$$\hat{W}_t = \mathcal{A}(W_{t-1}, x_t^e). \tag{3}$$

The candidate model with weights $\hat{W}_t$ is evaluated using its predictive distribution $P_{\hat{W}_t} : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the prompt and token spaces, respectively. For each probe x, the highest-probability token gives the point prediction:

$$\hat{y}_t(x) = \arg\max_{y \in \mathcal{Y}} P_{\hat{W}_t}(y \mid x). \tag{4}$$

However, repeated edits modify shared model parameters, allowing interference to accumulate if their effects are left unchecked. This can cause $\hat{y}_t(x)$ to differ from $y_t^*(x)$ and progressively degrade both edit success and preservation of reference behavior.

2.1 Joint Integrity–Utility Risk

This interference makes the effects of each candidate update difficult to anticipate. We therefore quantify the risk associated with this uncertainty by measuring exclusion of the evaluation targets $y_t^*(x)$ from prediction sets constructed using $P_{\hat{W}_t}$. Specifically, at probability-mass level $\beta \in [0, 1]$, we define the joint integrity–utility risk as:

$$D_t(\beta) = a \cdot \mathbb{E}_{x \in \mathcal{X}_t^{loc}} \left[L(y_{base}(x), C(x, \beta)) \right] + (1 - a) \cdot \mathbb{E}_{x \in \mathcal{X}_t^{in}} \left[L(o, C(x, \beta)) \right], \tag{5}$$

where $a \in [0, 1]$ balances integrity risk on locality probes against utility risk on generality probes, and each expectation is uniform over its corresponding finite probe set. Here, L is the miscoverage loss, which records whether a probe's evaluation target is excluded from the prediction set $C(x, \beta)$ or not. Specifically:

$$L(y, C(x, \beta)) = \mathbb{I}\big(y \notin C(x, \beta)\big). \tag{6}$$

To construct the prediction sets used in this loss, we define the edit boundary set predictor $C : \mathcal{X} \times [0, 1] \to 2^{\mathcal{Y}}$, where $2^{\mathcal{Y}}$ denotes the power set of the token space. For each probe x, we order tokens by decreasing probability under $P_{\hat{W}_t}$ and select the shortest prefix whose cumulative probability mass reaches β:

$$C(x, \beta) = \arg\min_{S} \left\{ |S| : \sum_{y \in S} P_{\hat{W}_t}(y \mid x) \geq \beta \right\}, \tag{7}$$

where S ranges over prefixes of this ordering. The boundary β specifies the prediction sets used to evaluate the fixed candidate weights $\hat{W}_t$. With this risk defined, our objective is to guide edit

[1] Construction details are provided in Appendix D.2.

admission using evidence about whether a candidate satisfies the risk criterion:

$$D_t(\beta) \le \theta, \tag{8}$$

where $\theta \in (0, 1)$ denotes the predefined tolerable joint integrity–utility risk. However, computing $D_t(\beta)$ exactly requires evaluating the candidate model on both probe sets at every step, making repeated assessment expensive.

2.2 Interference-Aware Sampled Risk Estimation

To address this efficiency challenge, we introduce an interference-aware sampling mechanism that uses a lightweight risk proxy to determine the probability of fully evaluating each candidate update. For each edit at time t, we define the interference tension score S_t as:

$$S_t = \frac{1}{k} \sum_{j \in \mathcal{N}_k(t)} \left\langle \frac{\phi_{\mathrm{w}}(x_t^e)}{\|\phi_{\mathrm{w}}(x_t^e)\|_2}, \frac{\phi_{\mathrm{w}}(x_j^e)}{\|\phi_{\mathrm{w}}(x_j^e)\|_2} \right\rangle, \tag{9}$$

where $\phi(\cdot)$ denotes a frozen sentence-embedding function and $\phi_{\mathrm{w}}(x) = (\Sigma_\phi + \varepsilon I)^{-1/2}(\phi(x) - \mu_\phi)$ is its covariance-corrected embedding, with $\varepsilon > 0$ a regularization constant and I the identity matrix. The statistics (μ_ϕ, Σ_ϕ) are estimated from a reference pool of edit prompts. The set $\mathcal{N}_k(t)$ contains the k nearest previously committed edits to x_t^e in this corrected embedding space. Thus, S_t summarizes similarity to nearby committed edits and serves as a proxy for potential interference.

We combine S_t with the evaluation gap $\Delta T_t = t - t_{\mathrm{last}}$, where t_{last} denotes the most recent full evaluation before step t, to obtain the proxy risk estimate:

$$\hat{r}_t = \sigma(w \cdot S_t + \gamma \cdot \Delta T_t + b). \tag{10}$$

Here $w, \gamma, b \in \mathbb{R}$ are learnable coefficients and σ is the sigmoid function. To prioritize evaluations for candidates with higher predicted risk, we define the full-evaluation probability as:

$$q_t = \min\left(1, \max\left(q_{\min}, \frac{\hat{r}_t}{\theta}\right)\right), \tag{11}$$

where $q_{\min} \in (0, 1]$ is the minimum evaluation probability. To select candidates for full probe evaluation, we sample a binary indicator $K_t \sim \mathrm{Bernoulli}(q_t)$. When $K_t = 1$, we evaluate both probe sets to obtain $D_t(\beta)$; otherwise, full probe evaluation is skipped. To account for this selective evaluation through the importance-sampled risk estimator, we define:

$$\tilde{R}_t(\beta) = \begin{cases} \hat{r}_t + \dfrac{D_t(\beta) - \hat{r}_t}{q_t}, & \text{if } K_t = 1, \\ \hat{r}_t, & \text{if } K_t = 0. \end{cases} \tag{12}$$

Here, $\hat{r}_t$ acts as a control variate, while the inverse-probability factor $1/q_t$ corrects for unequal evaluation probabilities. Having defined the sampled risk estimator, we next justify its use in the sequential monitor. The following lemma establishes that it preserves the conditional expected risk under selective evaluation:

Lemma 1. *Given $\tilde{R}_t(\beta)$ be the importance-sampled risk estimator Equation* (12)*, and let $\{\mathcal{F}_t\}_{t\ge 0}$ denote the filtration generated by the editing stream and the monitoring process. Assume that q_t and $\hat{r}_t$ are $\mathcal{F}_{t-1}$-measurable, $K_t \mid \mathcal{F}_{t-1} \sim \mathrm{Bernoulli}(q_t)$, and K_t is conditionally independent of $D_t(\beta)$ given $\mathcal{F}_{t-1}$. Then, for all t, we have: $\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big] = D_t(\beta)$. Consequently, if the null condition satisfies $\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \le \theta$ for all t, then:*

$$\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big] \le \theta.$$

Proof. Refer Appendix F.1. □

With conditional unbiasedness established, we next examine how the proxy improves estimation precision. The following theorem shows that, when the proxy is sufficiently accurate, the corrected risk estimator has conditional variance bounded above by that of the standard inverse-probability weighting (IPW) estimator under the same sampling probabilities:

Theorem 1. *Given $\tilde{R}_t(\beta)$ the importance-sampled risk estimator Equation* (12)*, let $R_t^{IPW}(\beta) = \frac{K_t}{q_t}\big(D_t(\beta)\big)$ denote the standard IPW estimator. Under the same measurability and sampling assumptions as Lemma 1, whenever the prior risk estimate $\hat{r}_t$ satisfies $\mathbb{E}\Big[\big(D_t(\beta) - \hat{r}_t\big)^2 \mid \mathcal{F}_{t-1}\Big] \le$*

$\mathbb{E}\big[D_t(\beta)^2 \mid \mathcal{F}_{t-1}\big]$, *the total conditional variance of the estimator is upper bounded by the variance of the standard IPW estimator lacking a control variate:*

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big) \le \mathrm{Var}\Big(R_t^{IPW}(\beta) \mid \mathcal{F}_{t-1}\Big).$$

Proof. Refer Appendix F.2. □

Remark 1. *Theorem 1 justifies using the proxy risk estimate $\hat{r}_t$ as a control variate for the sampled risk estimator. Under the stated prediction-error condition, the corrected estimator has no greater conditional variance than standard IPW at the same sampling probabilities. Thus, an accurate proxy can provide less variable risk estimates to SAVER's sequential monitor without additional full-probe evaluations.*

2.3 Editing Risk Martingale and Boundary Selection

The remaining challenge is to use sampled risk estimates to guide edit admission throughout the editing stream. Motivated by Ramdas et al. (2023), we convert $\tilde{R}_t(\beta)$ into an Edit Risk Martingale (ERM), denoted as $M_t(\beta)$, to obtain an anytime-valid false-alarm guarantee. It acts as a sequential monitor that accumulates evidence of excess joint integrity–utility risk. Specifically:

$$M_t(\beta) = \prod_{i=1}^{t} \Big(1 + \lambda_i^*(\beta)(\tilde{R}_i(\beta) - \theta)\Big), \tag{13}$$

where $\lambda_i^*(\beta) \in [0, \lambda_{\max}]$ is a sensitivity parameter with $\lambda_{\max} \coloneqq (\frac{1}{q_{\min}} - 1 + \theta)^{-1}$. By tracking it across $\mathcal{G} \subset [\beta_{\min}, \beta_{\max}]$ of size $m = |\mathcal{G}|$, we identify the *Knowledge Edit Boundary*, i.e., the smallest admissible value of β. Now, a fixed sensitivity may accumulate evidence inefficiently because the estimated risk varies across candidate boundaries. We therefore choose a boundary-specific sensitivity that maximizes cumulative log evidence on past observations:

$$\lambda_i^*(\beta) = \arg\max_{\lambda \in [0, \lambda_{\max}]} \sum_{j=1}^{i-1} \log\Big(1 + \lambda(\tilde{R}_j(\beta) - \theta)\Big). \tag{14}$$

Using this boundary-specific sensitivity in $M_t(\beta)$, we obtain one monitor value for each candidate boundary $\beta \in \mathcal{G}$. To control false alarms across time and these candidate boundaries, let $\alpha \in (0, 1)$ denote the global false-alarm level. We then select the smallest admissible knowledge-edit boundary $\hat{\beta}_t$, i.e., the minimum $\beta \in \mathcal{G}$ whose current monitor value is below the simultaneous rejection threshold, provided such a boundary exists. The objective is therefore formulated as:

$$\hat{\beta}_t = \inf \mathcal{B}_t, \quad \mathcal{B}_t \coloneqq \{\beta \in \mathcal{G} : M_t(\beta) < m/\alpha\} \tag{15}$$

where $\mathcal{B}_t$ denotes the feasible set in braces, and the threshold m/α accounts for the uniform allocation of α/m to each of the m candidate boundaries in $\mathcal{G}$. To characterize how the selected boundary relates to the oracle boundary, we provide the following conditional stability result:

Theorem 2. *Let $\tau \coloneqq m/\alpha$ and $\overline{\mathcal{B}}_t \coloneqq \{\beta \in [\beta_{\min}, \beta_{\max}] : M_t(\beta) < \tau\}$ and $\mathcal{B}_t^\star \coloneqq \{\beta \in [\beta_{\min}, \beta_{\max}] : M_t^\star(\beta) < \tau\}$ denote the continuous empirical and oracle feasible sets with $\bar{\beta}_t = \inf \overline{\mathcal{B}}_t$ and $\beta_t^\star = \inf \mathcal{B}_t^\star$ respectively. Assume both $\overline{\mathcal{B}}_t$ and $\mathcal{B}_t^\star$ are nonempty right-sided intervals and both monitors are strictly positive. Suppose there exist a uniform log-monitor error bound $\varepsilon_t > 0$, a local radius $\delta_{\mathrm{loc}} > 0$, and a local margin $m_t > 0$, describing how sharply the oracle monitor crosses the threshold, such that for all $\beta \in [\beta_{\min}, \beta_{\max}]$, $|\log M_t(\beta) - \log M_t^\star(\beta)| \le \varepsilon_t$, and for all $\beta \in [\beta_{\min}, \beta_{\max}]$ satisfying $|\beta - \beta_t^\star| \le \delta_{\mathrm{loc}}$, we have $\log M_t^\star(\beta) - \log \tau \ge m_t(\beta_t^\star - \beta)$ for $\beta < \beta_t^\star$, and $\log \tau - \log M_t^\star(\beta) \ge m_t(\beta - \beta_t^\star)$ for $\beta > \beta_t^\star$. If $\frac{\varepsilon_t}{m_t} < \delta_{\mathrm{loc}}$, we have:*

$$\big|\bar{\beta}_t - \beta_t^\star\big| \le \frac{\varepsilon_t}{m_t}.$$

Proof. Refer Appendix F.3. □

Remark 2. *Theorem 2 establishes conditional stability of boundary discovery after t sequential edits. Under the given conditions, the continuous empirical boundary $\bar{\beta}_t$ differ from its full-risk oracle counterpart by at most ε_t/m_t. Thus, selective evaluation can preserve the accuracy of boundary discovery when the cumulative monitor approximation remains sufficiently tight.*

To this end, we complete the modeling of the proposed SAVER framework. To produce admissible edit decisions, we design a pre-commit boundary discovery algorithm that selects the boundary

parameter $\hat{\beta}_t$. The complete SAVER procedure is summarized in Algorithm 1 in Appendix C and case study is provided in Appendix C.1.

Edit Admission. After processing a candidate edit e_t, SAVER obtains the feasible boundary set $\mathcal{B}_t$ from the boundary-specific monitors. If $\mathcal{B}_t$ is nonempty, the controller selects $\hat{\beta}_t = \inf \mathcal{B}_t$ as the admissible operating boundary and commits the candidate weights $\hat{W}_t$ as W_t. If $\mathcal{B}_t$ is empty, the candidate update is not applied and SAVER returns the last committed model W_{t-1}.

3 Theoretical Analysis

Next, we establish an anytime-valid false alarm control for SAVER and characterize its detection delay under persistent excess risk.

3.1 Anytime Valid False-Alarm Control

The following theorem establishes the anytime-valid guarantee for the ERM, ensuring the probability of a false alarm remains controlled across the interaction stream.

Theorem 3. *Given $M_t(\beta)$ be ERM with $\beta \in [\beta_{min}, \beta_{max}]$. Let $\{\mathcal{F}_t\}_{t\geq 0}$ denote all information available to the monitor up to time t. By property, $D_t(\beta)$ and $\hat{r}_t$ lie in $[0,1]$. Let the sampling rate $q_t \in [q_{\min}, 1]$ and predictable sensitivity $\lambda_t \in [0, \lambda_{\max}]$ be $\mathcal{F}_{t-1}$-measurable, and sampling indicator $K_t \mid \mathcal{F}_{t-1} \sim \text{Bernoulli}(q_t)$ be conditionally independent of $D_t(\beta)$. Under the null $\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \leq \theta$ for all t, the process $M_t(\beta)$ is a non-negative supermartingale with $\mathbb{E}[M_t(\beta)] \leq 1$. Consequently, for any $\alpha \in (0,1)$, it follows:*

$$\Pr\left(\exists t \in \mathbb{N} : M_t(\beta) \geq \alpha^{-1}\right) \leq \alpha.$$

Proof. Refer Appendix F.4. □

While Theorem 3 provides false-alarm control for a fixed boundary, SAVER evaluates multiple candidates in $\mathcal{G}$. Following result extends guarantee to simultaneous setting:

Corollary 1. *Let $\mathcal{G} \subset [\beta_{\min}, \beta_{\max}]$ be a finite grid of candidate coverage levels, and let $\{w_\beta\}_{\beta\in\mathcal{G}}$ be a set of positive weights satisfying $\sum_{\beta\in\mathcal{G}} w_\beta \leq 1$. Under the maintained-integrity null hypothesis, assume $\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \leq \theta$ holds for all $t \geq 1$ and all $\beta \in \mathcal{G}$. Assume the measurability and sampling conditions of Theorem 3 hold uniformly over $\mathcal{G}$. Then, the simultaneous probability of any false violation across all time steps and candidate boundaries is bounded by α:*

$$\Pr\left(\exists t \in \mathbb{N}, \exists \beta \in \mathcal{G} : M_t(\beta) \geq \frac{1}{\alpha w_\beta}\right) \leq \alpha.$$

Proof. Refer Appendix F.4.1 □

Remark 3. *Theorem 3 gives anytime-valid false-alarm control for a fixed boundary, while Corollary 1 extends it over the boundary grid used by* SAVER. *Thus, the admissible set of boundaries can be searched online without inflating the overall Type-I error. For uniform weights $w_\beta = 1/m$, the threshold $1/(\alpha w_\beta)$ reduces to m/α, which is the threshold used in Algorithm 1.*

3.2 Sequential Reliability under Excess Risk

We next bound the expected time for a fixed boundary monitor to signal persistent excess risk:

Theorem 4. *Let $\tau_{\alpha,t_0}(\beta) \coloneqq \inf\{t \geq t_0 : M_t(\beta) \geq \alpha^{-1}\}$ be post-change stopping time. Assume the defined monitoring conditions and the sampling rule in equation 11 hold, and $0 < M_{t_0-1}(\beta) < \alpha^{-1}$. Suppose there exist $t_0 \in \mathbb{N}$ and constants $\epsilon, \lambda_{\min}, \bar{\lambda}, \bar{v} > 0$ such that for all $t \geq t_0$: $\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \geq \theta + \epsilon, \quad \lambda_{\min} \leq \lambda_t \leq \bar{\lambda}$, and $\mathbb{E}[(\tilde{R}_t(\beta) - \theta)^2 \mid \mathcal{F}_{t-1}] \leq \bar{v}$. If $\bar{\lambda}\theta(2 - q_{\min}) \leq \frac{1}{2}$ and the drift $\delta_{\text{drift}} \coloneqq \lambda_{\min}\epsilon - \bar{\lambda}^2\bar{v} > 0$, then $\tau_{\alpha,t_0}(\beta) < \infty$ almost surely, and we have:*

$$\mathbb{E}[\tau_{\alpha,t_0}(\beta) - t_0 + 1 \mid \mathcal{F}_{t_0-1}] \leq \frac{\log \alpha^{-1} - \log M_{t_0-1}(\beta) + \log\left(1 + \bar{\lambda}\left(\frac{1}{q_{\min}} - \theta\right)\right)}{\delta_{\text{drift}}}.$$

Proof. Refer Appendix F.5. □

Finally, applying Markov's inequality to expected delay gives finite-horizon missed-detection bound:

Corollary 2. *Under the conditions of Theorem 4, for any horizon length $h \geq 1$,*

$$\Pr\left(\tau_{\alpha,t_0}(\beta) > t_0 + h - 1 \mid \mathcal{F}_{t_0-1}\right) \leq \frac{\log \alpha^{-1} - \log M_{t_0-1}(\beta) + \log\Big(1 + \bar{\lambda}\left(\frac{1}{q_{\min}} - \theta\right)\Big)}{\delta_{\text{drift}}\, h}.$$

Proof. Refer Appendix F.5.1. □

Remark 4. *Theorem 4 and Corollary 2 complement the anytime-valid false-alarm control in Theorem 3 by bounding expected detection delay and finite-horizon missed-detection probability under persistent excess risk.*

4 EXPERIMENTS

In this section, we evaluate SAVER through five research questions covering editing a) effectiveness, b) risk control, c) monitoring efficiency, d) robustness, and e) the contribution of its components. Specifically: **RQ1.** Can SAVER limit sequential degradation while retaining useful edits across editors, backbone LLMs, and edit-stream lengths? **RQ2.** How closely does the observed average probe-defined risk follow the user-specified tolerance? **RQ3.** Does interference-aware sampling reduce full-probe evaluation cost while retaining editing performance? **RQ4.** How does SAVER's performance depend on challenging edit streams and the coverage and quality of monitoring probes (Appendix E.3 and E.4)? **RQ5.** How do interference-aware sampling, adaptive boundary discovery, and the martingale monitor contribute to editing performance (Appendix E.5)? The hyperparameter analysis for conducting experiments is performed in Appendix E.6.

4.1 DATASETS AND BASELINE METHODS

Datasets and backbones. We evaluate SAVER on continuous edit streams constructed from CounterFact (Meng et al., 2022) and zsRE (Levy et al., 2017), with $T = 500$ submitted edits in the main comparisons. Main results use Qwen2.5-7B (Yang et al., 2024a), with additional backbone results on Llama-3.1-8B (Grattafiori et al., 2024). We also evaluate longer CounterFact streams of up to $T = 5000$ edits using Qwen3.5-9B (Team, 2026) with AlphaEdit. Stream and probe construction are detailed in Appendix D.2.

Editors and baselines. We instantiate SAVER-SAMPLED on ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), WISE (Wang et al., 2024), UltraEdit (Gu et al., 2026a), and AlphaEdit (Fang et al., 2024). We compare against adapted static and online conformal baselines, a fixed-β anytime-valid monitor, and current-probe-loss, KL-drift, and random-rejection gates matched to SAVER's acceptance rate. Within each setting, methods share the ordered edit requests, base-editor settings, and probe construction; each method generates candidate updates from its own last committed model. Baseline definitions are provided in Appendix D.4.

Evaluation metrics. We report Edit Success Rate (ESR) and Paraphrase Success Rate (PSR) to evaluate incorporation and generalization of the requested edits, and Neighborhood Success Rate (NSR) to measure preservation of reference behavior. Perplexity (PPL) on held-out text assesses general language-model quality beyond the editing probes. We also report acceptance rate and average probe-defined risk to assess edit admission, and evaluation rate and total runtime to measure computational cost. The matched-acceptance comparison additionally reports all-request ESR and PSR, counting rejected requests as failures. Metric definitions are provided in Appendix D.5. Implementation details, robustness stream construction, and parameter analyses appear in Appendices D.1, D.3 and E.6.

Across all editor–dataset pairs, SAVER improves NSR and reduces PPL, while also achieving higher ESR and PSR. ROME and MEMIT have the highest PPL after unconstrained editing, and both benefit from pre-commit control. For example, NSR increases from 0.029 to 0.059 for ROME on CounterFact and from 0.024 to 0.053 for MEMIT on zsRE, alongside reductions in PPL.

4.1.1 RESULTS W.R.T. LONG-RUN EDITING WITH EXISTING EDITORS

We compare each base editor with its SAVER-Sampled variant over $T = 500$ submitted edits. Table 1 reports results on Qwen2.5-7B, with corresponding Llama-3.1-8B results in Appendix E.1.

The improvements also extend to editors designed for preservation or lifelong editing. For AlphaEdit and UltraEdit, SAVER reduces PPL by 16.1% and 10.9% on CounterFact, respectively. Across the two datasets, it commits 97.8–99.2% of their submitted updates. The hidden-representation visualization in Figure 3 also shows a smaller pre/post distribution shift for AlphaEdit with SAVER.

Table 1: Long-run editing with existing editors on **CounterFact** and **zsRE** with and w/o SAVER for **Qwen2.5-7B** in terms of Acc., ESR, PSR, NSR, and PPL over 500 submitted candidate updates. Parentheses show SAVER's relative change over the unconstrained editor. Results are averaged over 5 seeds; $p < 0.05$ by a paired two-sided test vs the best competing method.

Editor	Method	CounterFact					zsRE				
		Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓
ROME	Unconstr.	1.000	0.192	0.115	0.029	100.97	1.000	0.172	0.179	0.019	87.86
	+SAVER	0.882	0.225 (+17.2%)	0.126 (+9.6%)	0.059 (+103.4%)	89.96 (-10.9%)	0.892	0.192 (+11.6%)	0.184 (+2.8%)	0.030 (+57.9%)	80.79 (-8.0%)
MEMIT	Unconstr.	1.000	0.217	0.123	0.045	89.51	1.000	0.198	0.192	0.024	81.86
	+SAVER	0.908	0.223 (+2.8%)	0.153 (+24.4%)	0.064 (+42.2%)	81.14 (-9.4%)	0.920	0.229 (+15.7%)	0.209 (+8.9%)	0.053 (+120.8%)	75.91 (-7.3%)
WISE	Unconstr.	1.000	0.234	0.217	0.075	57.51	1.000	0.188	0.196	0.064	61.86
	+SAVER	0.968	0.285 (+21.8%)	0.223 (+2.8%)	0.104 (+38.7%)	51.14 (-11.1%)	0.950	0.234 (+24.5%)	0.219 (+11.7%)	0.091 (+42.2%)	55.91 (-9.6%)
UltraEdit	Unconstr.	1.000	0.372	0.315	0.059	45.97	1.000	0.382	0.259	0.049	43.86
	+SAVER	0.992	0.425 (+14.2%)	0.344 (+9.2%)	0.089 (+50.8%)	40.96 (-10.9%)	0.982	0.422 (+10.5%)	0.294 (+13.5%)	0.061 (+24.5%)	40.79 (-7.0%)
AlphaEdit	Unconstr.	1.000	0.844	0.590	0.126	49.64	1.000	0.724	0.526	0.033	48.86
	+SAVER	0.986	0.878 (+4.0%)	0.627 (+6.3%)	0.157 (+24.6%)	41.67 (-16.1%)	0.978	0.746 (+3.0%)	0.543 (+3.2%)	0.038 (+15.2%)	42.61 (-12.8%)

The Llama-3.1-8B results show similar preservation gains (Appendix E.1). At comparable acceptance rates, SAVER achieves higher NSR and lower PPL than the current-probe-loss, KL-drift, and random-rejection gates (Appendix E.1.2). This comparison further supports the effectiveness of its admission decisions. The benefits persist over longer streams with Qwen3.5-9B + AlphaEdit (Appendix E.1.3). At 5,000 submitted edits, SAVER commits 95.2% of updates while reducing PPL from 162.7 to 76.4 and improving NSR from 0.044 to 0.102. ESR also increases from 0.574 to 0.731. The performance gap widens as the stream grows, showing that the benefits remain substantial over sequences ten times longer than the main evaluation setting.

4.1.2 Results w.r.t. User-Specified Risk Control

We next compare SAVER-Sampled with representative monitoring baselines in terms of user-specified risk control and editing utility. We use Qwen2.5-7B + AlphaEdit with $\theta = 0.10$ and $\alpha = 0.10$, and report the average probe-defined joint risk $\widehat{D}$ alongside acceptance, edit utility, and preservation metrics.

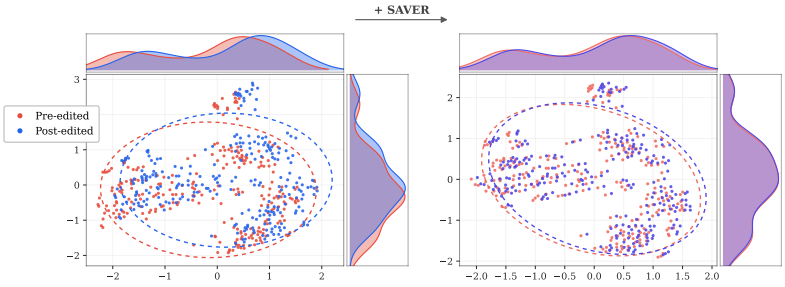


Figure 3: Hidden representations before and after 500 edits on zsRE using AlphaEdit. With SAVER, the post-edit distribution remains closer to the pre-edit reference.

As shown in Table 2, SAVER achieves average risk below the requested threshold on both datasets: $\widehat{D} = 0.095$ on CounterFact and $\widehat{D} = 0.090$ on zsRE. The corresponding averages exceed θ for static conformal, online conformal, and fixed-β monitoring. Online conformal lowers risk relative to static calibration, while fixed-β monitoring achieves higher NSR and lower PPL than both conformal baselines. However, SAVER further improves these preservation metrics and achieves higher ESR and PSR.

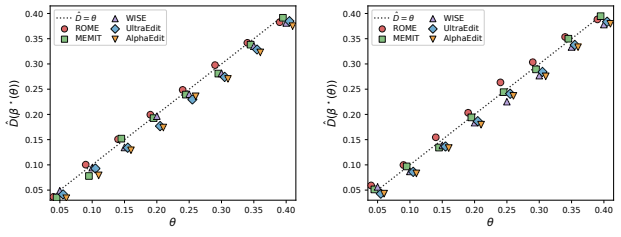
Table 2: Monitoring baselines using Qwen2.5-7B + AlphaEdit at $\alpha = 0.10$ and $\theta = 0.10$. † marks average joint risk $\widehat{D} > \theta$. Results are averaged over five seeds.

Dataset	Monitor	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	$\widehat{D}\downarrow$
CounterFact	Static conformal	0.798	0.801	0.585	0.055	55.14	0.160†
	Online conformal	0.825	0.822	0.592	0.063	49.85	0.130†
	Fixed β	0.806	0.844	0.601	0.139	46.18	0.130†
	SAVER	**0.986**	**0.878**	**0.627**	**0.157**	**41.67**	**0.095**
zsRE	Static conformal	0.756	0.732	0.496	0.018	53.89	0.170†
	Online conformal	0.812	0.729	0.509	0.021	47.05	0.101†
	Fixed β	0.768	0.735	0.531	0.034	45.51	0.140†
	SAVER	**0.978**	**0.746**	**0.543**	**0.038**	**42.61**	**0.090**

We further vary θ from 0.05 to 0.40 across the five editors. In Figure 4, achieved risk broadly follows the requested tolerance on both datasets, with differences across editors. Smaller thresholds lead to lower observed joint risk.

The additional probe analyses examine how monitoring coverage and relevance affect preservation on disjoint locality probes (Appendix E.4). Increasing monitoring coverage from 25% to 100% reduces held-out locality miscoverage from 0.121 to 0.095. Replacing the monitoring locality prompts

with unrelated prompts instead raises held-out miscoverage to 0.124 at $T = 500$, highlighting the importance of the behaviors represented by the monitoring probes.



(a) CounterFact (b) zsRE

Figure 4: Average joint risk under varying thresholds θ across five SAVER-wrapped editors. The dotted line denotes $\widehat{D} = \theta$.

Finally, a controlled experiment examines detection under persistent excess risk with full-probe evaluation (Appendix E.2.1). As excess risk increases from 0.037 to 0.233, the median detection delay at a fixed boundary decreases from 400 to 64 monitoring steps, showing a faster response to larger sustained violations.

4.1.3 Results w.r.t. Interference-Aware Monitoring Efficiency

We next evaluate whether interference-aware sampling reduces monitoring cost while maintaining edit utility and preservation of reference behavior.

Table 3: Monitoring efficiency on **CounterFact** and **zsRE** using Qwen2.5-7B + AlphaEdit over 500 submitted edits. Results are averaged over five seeds. Best values among monitoring variants are in **bold**; second best are underlined.

Dataset	Method	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	Sample Rate↓	Time (h)↓	Var.↓
CounterFact	Unconstr.	1.000	0.844	0.590	0.126	49.64	–	2.68	–
	SAVER-Full	**0.987**	**0.886**	0.625	**0.157**	41.78	1.000	3.21	**0.0613**
	NoProxy	0.976	0.856	0.580	0.152	51.50	**0.770**	**2.71**	0.0965
	SAVER-Sampled	0.986	0.878	**0.627**	**0.157**	**41.67**	0.774	2.73	0.0713
zsRE	Unconstr.	1.000	0.724	0.526	0.033	48.86	–	1.23	–
	SAVER-Full	0.967	0.738	**0.548**	**0.041**	**42.20**	1.000	2.18	**0.0487**
	NoProxy	0.942	0.609	0.423	0.021	46.73	0.780	**1.32**	0.1059
	SAVER-Sampled	**0.978**	**0.746**	0.543	0.038	42.61	**0.754**	1.33	0.0628

SAVER-Full evaluates the complete online probe bank for every candidate and serves as the full-evaluation reference. SAVER-Sampled uses semantic-interference and temporal-drift signals to prioritize full-probe evaluation and construct the proxy correction. NoProxy instead uses constant sampling probability and non-adaptive prior, with its expected evaluation budget matched to SAVER-Sampled. As shown in Table 3, SAVER-Full improves editing utility and preservation relative to unconstrained editing, but requires additional computation. SAVER-Sampled maintains comparable performance while reducing full-probe evaluations by 22.6% on CounterFact and 24.6% on zsRE. Total runtime decreases from 3.21 to 2.73 hours and from 2.18 to 1.33 hours, respectively, corresponding to reductions of 15.0% and 39.0% relative to SAVER-Full. Its runtime is 1.9% and 8.1% above unconstrained editing on the two datasets. On CounterFact, SAVER-Sampled matches SAVER-Full on NSR, with slightly higher PSR and lower PPL, although ESR is slightly lower. On zsRE, it admits more updates and achieves higher ESR, while PSR, NSR, and PPL remain close to full-evaluation reference. Thus, the reduction in evaluation cost retains much of the editing performance obtained through exhaustive monitoring.

At comparable realized sample rates, NoProxy is slightly faster but achieves lower ESR, PSR, and NSR and higher PPL than SAVER-Sampled on both datasets. The difference is especially pronounced on zsRE, where ESR falls from 0.746 to 0.609 and NSR from 0.038 to 0.021 without the adaptive sampling and proxy correction. The empirical variance of logged risk estimates is also lower with SAVER-Sampled: 0.0713 versus 0.0965 on CounterFact and 0.0628 versus 0.1059 on zsRE. These results support the combined use of interference-aware evaluation allocation and proxy correction to retain editing performance at a reduced evaluation cost.

5 Conclusion

This paper investigates the underexplored issue of reliability in sequential KE, where repeated updates may degrade LLM behaviors. We integrate edit utility and reference-behavior preservation into a unified pre-commit control framework called SAVER, which dynamically selects knowledge-edit boundaries and commits updates only when specified probe-defined risk threshold can be satisfied. Our theoretical analysis and extensive experiments validate the effectiveness and efficiency of SAVER. It is noteworthy that the SAVER depends on the quality of the probe, thus designing richer probes remains an important direction for future work. Overall, SAVER can work agnostically on any knowledge editing method by treating the editor as a black box, offering a statistically grounded reliability layer for more trustworthy continual model editing.

AI USE STATEMENT

We used an AI-assisted writing tool (Grammarly), to polish the manuscript's language and correct grammatical errors and typos.

REPRODUCIBILITY STATEMENT

Implementation details and experimental settings are provided in Appendix D.1. The theoretical results are accompanied by proofs and remarks clarifying their assumptions and implications. The code repository provided includes data preprocessing procedures and scripts for running the experiments and generating tables and figures.

REFERENCES

Anastasios Nikolas Angelopoulos, Rina Barber, and Stephen Bates. Online conformal prediction with decaying step sizes. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1616–1630. PMLR, 21–27 Jul 2024a.

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024b.

Stephen Bates, Anastasios Nikolas Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *J. ACM*, 68:43:1–43:34, 2021.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Peter Grünwald, Rianne de Heide, and Wouter M Koolen. Safe testing. In *2020 Information theory and applications workshop (ITA)*, pp. 1–54. IEEE, 2020.

Xiaojie Gu, Ziying Huang, Jia-Chen Gu, and Kai Zhang. Ultraedit: Training-, subject-, and memory-free lifelong editing in language models. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856.

Zihan Gu, TianYi Zhang, Xinyan Zhang, Zhiyuan Wang, Han Zhang, Yuhao Wei, Jiacheng Lu, Tianyi Ma, Xingsheng Zhang, Hua Zhang, and Yue Hu. Diagnosing hidden instabilities in model editing via uncertainty quantification. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32544–32566, San Diego, California, United States, July 2026b. Association for Computational Linguistics.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2), April 2021. ISSN 0090-5364.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, 2017.

Qi Li and Xiaowen Chu. Can we continually edit language models? on the knowledge attenuation in sequential model editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5438–5455, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Qi Li and Xiaowen Chu. AdaEdit: Advancing continuous knowledge editing for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria, July 2025. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9: 371–421, June 2008. ISSN 1532-4435.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3), 2021. ISSN 0090-5364.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. 01 2005.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: rethinking the knowledge memory for lifelong model editing of large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5419–5437, Bangkok, Thailand, August 2024b. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, Singapore, December 2023. Association for Computational Linguistics.

INDEX

- Related Works . . . . . p. 12
- Notations Table . . . . . p. 13
- Algorithm . . . . . p. 14
- Case Study . . . . . p. 14
- Experimental Details . . . . . p. 15
- Additional Results . . . . . p. 21
- Proofs . . . . . p. 25

A RELATED WORKS

A.1 KNOWLEDGE EDITING

Knowledge editing (KE) aims to modify factual behavior in large language models without full retraining. Early parameter-editing methods primarily focus on locating and updating the internal components responsible for factual recall. For example, ROME edits localized factual associations through rank-one updates, while MEMIT extends this idea to mass editing by writing many associations into transformer memories (Meng et al., 2022; 2023). Other works formulate editing through learned or memory-based mechanisms, such as SERAC (Mitchell et al., 2022), which stores edits externally and modulates model predictions through a retrieval-augmented counterfactual model. These methods have substantially advanced single-edit and batch-edit efficacy, but they often have lesser performance in continual editing setting. Subsequently, recent work has begun to address continual and lifelong editing, where many updates are applied sequentially. WISE (Wang et al., 2024) rethinks edited knowledge as a memory management problem and introduces a dual-memory mechanism for lifelong model editing. AlphaEdit constrains parameter perturbations through preserved-knowledge geometry to reduce disruption during editing (Fang et al., 2024). UltraEdit targets scalable lifelong editing with a training, subject and memory-free update mechanism (Gu et al., 2026a), and AdaEdit further studies continuous editing under large-scale sequential updates (Li & Chu, 2025). These methods reduce interference by changing how edits are written. In parallel, there have been works studies on how edited models fail beyond direct edit success. MQuAKE shows that edited models often fail to propagate changes to multi-hop consequences (Zhong et al., 2023), while RippleEdits evaluates whether a local edit induces consistent changes in related facts (Cohen et al., 2024). Sequential editing introduces further aggravation of reliable editing. Knowledge attenuation describes how previously edited knowledge can decay as more edits accumulate (Li & Chu, 2024), and the butterfly-effect analysis shows that even few edits can trigger broad model degradation (Yang et al., 2024b). However, these works are primarily diagnostic or editor-side and donot provide any grounded mechanism to address it. Addressing this gap is the focus of our work.

A.2 RISK-CONTROLLED PREDICTION SETS AND ANYTIME-VALID MONITORING

Our work is also motivated by the uncertainty quantification (Vovk et al., 2005; Vovk & Wang, 2021; Howard et al., 2021; Ramdas et al., 2023) and risk-controlled prediction (Bates et al., 2021; Angelopoulos et al., 2024b). Conformal prediction provides distribution-free tools for producing sets with finite-sample validity (Shafer & Vovk, 2008; Vovk et al., 2005). Risk-controlling prediction sets (RCPS) extend this idea by allowing users to specify a loss function and construct prediction sets whose expected risk is controlled below a target threshold (Bates et al., 2021; Angelopoulos et al., 2024b). These frameworks have been applied in settings where the model output can be converted into a set-valued prediction and evaluated against observed labels or losses. However, standard risk-control methods are not directly designed for sequential editing, where accepting a candidate update changes the model that generates future outputs and future risks. Online conformal prediction extends conformal ideas to sequential observations by updating calibration rules as data arrive (Angelopoulos et al., 2024a). In parallel, safe testing and anytime-valid inference develop evidence processes and martingale-based tests that preserve Type-I error control under optional stopping or continuous monitoring (Grünwald et al., 2020; Ramdas et al., 2023). These methods provide the statistical language for sequential validity, but their natural setting is a prediction or testing stream in which monitoring observes feedback without rewriting the predictor whose future outputs are evaluated. Sequential KE differs from these methods as the monitored decision is itself a parameter

intervention, and the relevant risk is not directly observed. Our research paper aims to apply these ideas in sequential editing to address these limitations.

B NOTATIONS TABLE

Here we provide the notations used in the paper.

Table 4: Summary of Notations

Notation	Description
$\mathcal{W}$	Parameter space of the language model.
W_0	Initial pre-trained model weights before the edit stream.
W_t	Last committed model weights after processing edit step t.
$\widehat{W}_t$	Candidate model weights proposed by the base editor before commit.
$\mathcal{E} = \{e_t\}_{t=1}^{T}$	Sequential stream of submitted candidate edits.
T	Length of the edit stream; in experiments, $T = 500$.
$e_t = (s, r, o)$	Candidate factual edit at step t, with subject s, relation r, and target object o.
$\mathcal{A}$	Base editing algorithm, e.g., ROME, MEMIT, WISE, UltraEdit, or AlphaEdit.
$\mathcal{T}$	Submitted edit stream used in evaluation.
$\mathcal{C} \subseteq \mathcal{T}$	Set of candidate updates committed by a method.
$\mathcal{I}$	Probe generator that maps an edit triple to an edit prompt and probe sets.
x_t^e	Edit prompt generated for candidate edit e_t.
$\mathcal{X}_t^{\text{in}}$	In-scope generality probe set for edit e_t.
$\mathcal{X}_t^{\text{loc}}$	Locality probe set for edit e_t.
$y_t^*(x)$	Reference target associated with probe x.
$y_{\text{base}}(x)$	Reference token induced by the initial model W_0 on locality probe x.
$y_{\text{ref}}(x)$	Chosen reference behavior used for locality evaluation.
$\mathcal{X}, \mathcal{Y}$	Prompt space and token space.
$P_{\widehat{W}}(y \mid x)$	Predictive probability of token y under candidate model $\widehat{W}_t$.
$C(x, \beta)$	Set-valued output whose cumulative predictive mass reaches boundary β.
β	Boundary parameter controlling prediction-set size and conservativeness.
$\mathcal{G}_\beta$ or $\mathcal{G}$	Candidate boundary grid searched by SAVER.
$m = \lvert\mathcal{G}\rvert$	Number of candidate boundaries in the grid.
$L(y, C(x, \beta))$	Miscoverage loss, $L(y, C(x, \beta)) = \mathbb{I}[y \notin C(x, \beta)]$.
a	Trade-off weight between locality/reference preservation and in-scope edit utility.
$D_t(\beta)$	Theoretical joint probe-defined post-edit distortion risk at step t.
$\widehat{D}_t(\beta)$	Empirical full-probe estimate of $D_t(\beta)$ used for reporting.
$\widehat{D}(\beta)$	Stream-level post-hoc distortion risk averaged over committed updates.
θ	User-specified post-edit distortion risk threshold.
α	Sequential significance level controlling false-alarm probability.
$\phi(\cdot)$	Frozen sentence-embedding function used to compute edit similarity.
$\phi_{\text{w}}(x)$	Covariance-corrected embedding of prompt x.
μ_ϕ, Σ_ϕ	Mean and covariance estimated from a reference pool of edit prompts.
$\mathcal{N}_k(t)$	Set of k-nearest previously committed edits to x_t^e.
S_t	interference tension score measuring local edit-neighborhood crowding.
$\Delta T_t = t - t_{\text{last}}$	Temporal drift since the most recent full-probe evaluation.
t_{last}	Most recent time step at which full-probe evaluation was performed.
$\widehat{r}_t$	interference prior risk estimate from semantic interference and temporal drift.

Notation	Description
w, γ, b	Learnable coefficients used in the interference risk prior.
$\sigma(\cdot)$	Sigmoid activation used to map the risk-prior score into $[0, 1]$.
K_t	Binary full-evaluation indicator, $K_t \sim \mathrm{Bernoulli}(q_t)$.
q_t	Sampling probability for full-probe evaluation at step t.
$q_{\min}$	Minimum full-evaluation probability.
$\widetilde{R}_t(\beta)$	Control-variate importance-sampled estimator of $D_t(\beta)$.
$R_t^{\mathrm{IPW}}(\beta)$	Standard inverse-propensity weighted risk estimator without the control variate.
$\mathcal{F}_t$	Monitoring filtration containing committed edit history and pre-sampling candidate features up to time t.
$M_t(\beta)$	Edit Risk Martingale/ERM monitor for boundary β.
λ_i	Predictable sensitivity parameter used in the ERM update at step i.
$\lambda_{\max}$	Maximum admissible sensitivity ensuring non-negative ERM factors.
$\lambda_i^*(\beta)$	Boundary-specific sensitivity maximizing cumulative past evidence growth.
w_β	Grid weight assigned to boundary β for simultaneous validity.
m/α	Uniform-grid rejection threshold when $w_\beta = 1/m$.
$\mathcal{B}_t$	Empirical feasible boundary set at time t, $\{\beta : M_t(\beta) < m/\alpha\}$.
$\widehat{\beta}_t$	Smallest admissible boundary selected by SAVER, $\widehat{\beta}_t = \inf \mathcal{B}_t$.
$M_t^\star(\beta)$	Oracle monitor computed from the oracle/full-risk sequence.
$\mathcal{B}_t^\star$	Oracle feasible boundary set.
$\beta_t^\star$	Oracle boundary, $\beta_t^\star = \inf \mathcal{B}_t^\star$.
τ	Rejection threshold used in the boundary-localization theorem; under uniform grid, $\tau = m/\alpha$.
ε_t	Uniform log-monitor approximation error between empirical and oracle monitors.
m_t	Local margin controlling sharpness of the oracle boundary.
δ_{loc}	Local radius around the oracle boundary used in boundary localization.
$\tau_{\alpha,t_0}(\beta)$	Post-change stopping time for monitor $M_t(\beta)$.
ϵ	Persistent excess risk above threshold θ.
$\lambda_{\min}, \bar{\lambda}$	Lower and upper bounds on predictable sensitivity in the finite-time halting theorem.
$\bar{v}$	Upper bound on the conditional second moment of $\widetilde{R}_t(\beta) - \theta$.
δ_{drift}	Expected log-growth (drift) of the monitor under persistent excess risk, $\lambda_{\min}\epsilon - \bar{\lambda}^2\bar{v}$.
Acc.	Acceptance rate, the fraction of submitted candidate updates committed by the method.
ESR	Edit Success Rate on the submitted edit prompt.
PSR	Paraphrase Success Rate on in-scope generality probes.
NSR	Neighborhood Success Rate on locality/reference probes.
PPL	Perplexity on the held-out reference corpus.
Sample Rate	Fraction of candidate updates receiving full-probe evaluation.
Time	Wall-clock runtime required to process the edit stream.
Var.	Empirical variance of the sampled risk estimator over the stream.

C ALGORITHM

Here, we provide the SAVER algorithm, which helps us in selecting the boundary parameter $\hat{\beta}_t$ for save knowledge editing in LLM.

C.1 CASE STUDY: EDIT ADMISSION AND REJECTION

We examine edit requests from the prepared CounterFact and zsRE streams. The CounterFact request changes the country associated with St John's Church, Kingston upon Thames, while its lo-

Algorithm 1 The SAVER Algorithm

Input: Initial model W_0, edit stream $\{e_t\}_{t=1}^{T}$, risk threshold θ, significance α, boundary grid $\mathcal{G} \subset [\beta_{\min}, \beta_{\max}]$ with $m = |\mathcal{G}|$
Output: Last committed model W_t, selected admissible boundary $\hat{\beta}_t$
Initialize: $M_0(\beta) = 1$ for all $\beta \in \mathcal{G}$, $\hat{\beta}_0 = \beta_{\min}$, $t_{\text{last}} = 0$
for $t = 1, 2, \ldots, T$ **do**
    Generate prompt x_t^e and probe sets $\mathcal{X}_t^{in}, \mathcal{X}_t^{loc}$ via $I(s, r, o)$
    Propose candidate weights: $\hat{W}_t = \mathcal{A}(W_{t-1}, x_t^e)$
    Compute edit interference tension S_t using eq. (9)
    Set temporal drift $\Delta T_t = t - t_{\text{last}}$
    Compute the prior risk estimate $\hat{r}_t$ using eq. (10)
    Compute the sampling rate q_t using eq. (11)
    Sample K_t conditionally as $K_t \sim \text{Bernoulli}(q_t)$
    if $K_t = 1$ **then**
        Compute the joint risk $D_t(\beta)$ for all $\beta \in \mathcal{G}$ using eq. (5)
        $t_{\text{last}} \leftarrow t$
        Compute $\tilde{R}_t(\beta)$ for all $\beta \in \mathcal{G}$ using eq. (12)
    else
        Set $\tilde{R}_t(\beta) \leftarrow \hat{r}_t$ for all $\beta \in \mathcal{G}$
    end if
    For each $\beta \in \mathcal{G}$, compute $\lambda_t^*(\beta) = \arg\max_{\lambda \in [0, \lambda_{\max}]} \sum_{j=1}^{t-1} \log\left(1 + \lambda(\tilde{R}_j(\beta) - \theta)\right)$
    Update $M_t(\beta)$ for all $\beta \in \mathcal{G}$ using eq. (13)
    $\mathcal{B}_t \leftarrow \{\beta \in \mathcal{G} : M_t(\beta) < m/\alpha\}$
    if $\mathcal{B}_t \neq \emptyset$ **then**
        Solve for $\hat{\beta}_t = \inf \mathcal{B}_t$ using eq. (15)
        Commit updated weights: $W_t \leftarrow \hat{W}_t$
    else
        Reject candidate update: $W_t \leftarrow W_{t-1}$
    end if
end for

cality probe concerns religious affiliation. The zsRE request changes the record label of *Runaway Sunday*, while its locality probe concerns distribution format. Figure 5 presents prompt excerpts from these records and illustrates the acceptance and rejection branches of Algorithm 1.

Acceptance. For the church edit, the in-scope generality probe tests the requested country association through an alternative name for the same subject. The locality probe asks about religious affiliation, a relation outside the requested change. SAVER combines their reference-token miscoverage with the remaining probes in the joint risk and updates the sequential monitor. If at least one boundary remains admissible, Eq. (15) selects the smallest such boundary and the candidate weights are committed. The selected boundary determines the prediction-set coverage used by the controller while the proposed weights remain fixed.

Rejection. For the album edit, support for the requested label *Motown* is evaluated alongside the locality reference *music streaming*. The requested label can be supported while the locality reference loses support. If the accumulated monitoring evidence leaves no admissible boundary in G, SAVER rejects the candidate and retains the last committed model. The stream then proceeds to the next edit.

These cases distinguish the factual relation being updated from the reference behavior being preserved. Admission depends on the accumulated evidence in $M_t(\beta)$ and does not certify the individual candidate's risk.

D Experimental Details

This appendix provides details needed to reproduce the experiments in Section 4. We describe implementation choices, edit-stream construction, monitoring baselines, and evaluation metrics.

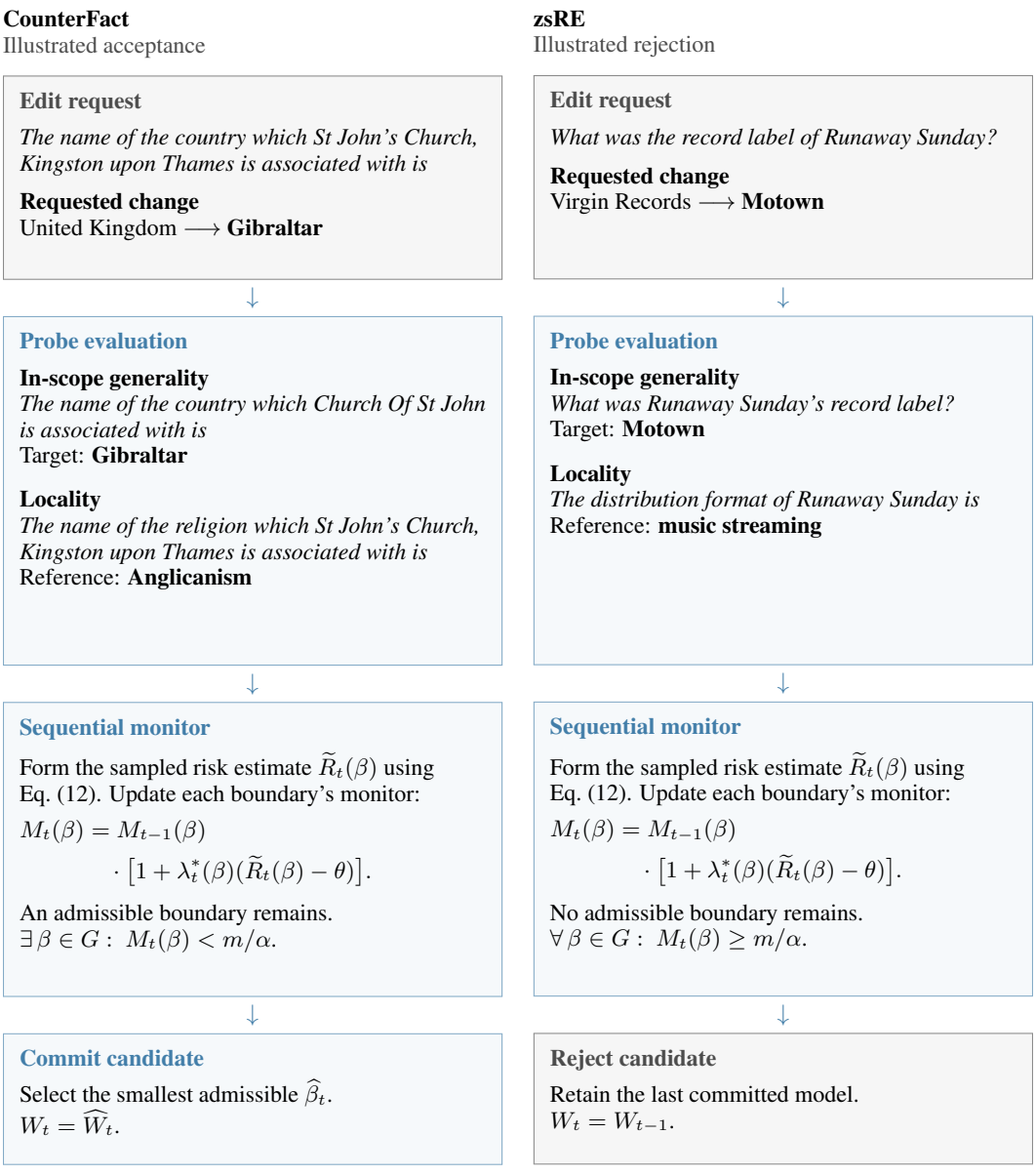


Figure 5: Case study showing SAVER's edit admission and rejection using CounterFact and zsRE examples. The schematics trace each edit request through probe evaluation and sequential monitoring to the commit or reject decision.

D.1 Implementation Details

We use continuous edit streams of length $T = 500$ for the main experiments and report results averaged over five random seeds. The main experiments use $\alpha = 0.10$ as the sequential significance level and $\theta = 0.10$ as the default joint integrity–utility risk threshold, with trade-off weight $a = 0.7$. For the threshold analysis in RQ2, we sweep $\theta \in \{0.05, 0.10, \ldots, 0.40\}$. For sampled SAVER, the minimum full-evaluation probability is set to $q_{\min} = 0.25$. We use the boundary grid $\mathcal{G} = \{0.55, 0.57, \ldots, 0.99\}$, containing $m = 23$ candidate boundaries with $\beta_{\max} = 0.99$. We compute q_t and $\hat{r}_t$ from the committed edit history and lightweight candidate features, including semantic-overlap and temporal-drift signals, before drawing K_t. We further use Qwen2.5-7B and Llama-3.1-8B as backbone LLMs. The editor-wrapper experiments instantiate SAVER on ROME, MEMIT, WISE, UltraEdit, and AlphaEdit. For the monitoring and ablation setting, we use Qwen2.5-7B + AlphaEdit. For the long-stream analysis, we use Qwen3.5-9B + AlphaEdit on CounterFact with $T \in \{100, 500, 1000, 2500, 5000\}$. For each editor, we keep hyperparameters fixed between the unconstrained editor and its SAVER-wrapped variant. All experiments are implemented using the EasyEdit codebase with Python 3.11.7, PyTorch 2.2.2, Transformers 4.57.1, NumPy 1.26.4, SciPy 1.17.1, PEFT 0.19.1, and SentencePiece 0.2.1. Experiments are run on NVIDIA H100 NVL GPUs and NVIDIA A40 GPUs.

D.2 Datasets and Edit-Stream Construction

We use CounterFact and zsRE to construct ordered streams of factual edits. For each seed, all methods receive the same edit order and dataset-provided prompts. Each method applies the base editor to its last committed model.

Probe construction. For an edit $e_t = (s, r, o)$, the probe generator $\mathcal{I}$ constructs the edit prompt x_t^e, in-scope generality probes $\mathcal{X}_t^{in}$, and locality probes $\mathcal{X}_t^{loc}$ using the corresponding dataset record. In CounterFact, the edit prompt is obtained by substituting the subject s into the supplied relation template where required. In zsRE, the supplied source question serves as the edit prompt. The new target object o accompanies this prompt when the base editor proposes candidate weights $\hat{W}_t$.

In-scope generality probes assess support for the edited fact across equivalent formulations. Dataset rephrases supply these prompts, and the edit prompt itself is included in $\mathcal{X}_t^{in}$. Locality probes are drawn from the dataset's neighborhood or locality questions. For locality monitoring, the reference token $y_{base}(x)$ is the token predicted by the initial model W_0 on probe x. Surrounding whitespace is stripped, and empty or duplicate paraphrase strings are removed during dataset conversion. Each retained probe is paired with its evaluation target.

Prompt representation and evaluation. CounterFact probes are factual cloze prompts, while zsRE probes are natural-language questions. The prompt text is tokenized directly as input to the causal language model, and its evaluation target is stored separately. For in-scope generality probes, the reference token is the first token of the space-prefixed target object o, tokenized without special tokens. The candidate model's next-token distribution at the end of the prompt is used to construct $C(x, \beta)$. Reference-token miscoverage on locality and in-scope generality probes is combined into the joint integrity–utility risk in Eq. 5.

Online monitoring and held-out evaluation. At round t, SAVER samples $K_t \sim \text{Bernoulli}(q_t)$. When $K_t = 1$, it evaluates the online in-scope generality and locality probes across the boundary grid. When $K_t = 0$, it uses the proxy risk estimate. Eq. 12 defines the resulting sampled risk estimator.

For the probe-coverage and probe-quality analyses, the locality probes $\mathcal{X}_t^{loc}$ are divided into a monitoring subset and a disjoint held-out audit subset. For edits with sufficient locality probes, a shuffle determined by the experiment seed and edit identifier partitions the probe pairs as evenly as possible before editing. The monitoring subset is used together with $\mathcal{X}_t^{in}$ for online risk evaluation. The held-out audit subset is fixed across conditions, and its losses are excluded from the proxy risk estimate, sampling probability, sequential monitor, and boundary selection.

Probe-coverage sensitivity varies the number of locality probes selected from the monitoring subset, with smaller selections nested within larger ones. The in-scope generality probes and held-out audit subset remain fixed. Probe-quality sensitivity replaces the monitoring locality prompts with donor locality prompts from other edits, selected using prompt similarity and prompt and answer lengths.

Each donor probe uses the reference token predicted by W_0 on its own prompt. Held-out evaluation reports locality miscoverage and Neighborhood Success Rate (NSR). Spearman correlation is computed from paired monitored and held-out risks on rounds for which both are evaluated.

D.3 Robustness Stream Construction

We construct four controlled edit-stream settings to evaluate whether SAVER remains effective when sequential interference is amplified. All robustness streams contain $T = 500$ submitted edits from CounterFact and use Qwen2.5-7B + AlphaEdit, with the same probe-construction pipeline and evaluation metrics as the main experiments.

- **Overlap burst.** This setting tests what happens when many nearby edits arrive in a short burst. Starting from the shuffled CounterFact stream, we compute an overlap score between candidate edits using token and relation overlap. We then select anchor edits and place several high-overlap neighboring edits immediately after each anchor. This creates local clusters of semantically or relation-wise similar updates. The setting stresses correlated locality damage, where several nearby edits can perturb the same reference neighborhood.
- **Contradictory re-edit loop.** This setting tests direct overwrite pressure on the same underlying fact prompt. We choose anchor edits and construct four-step loops of the form
$$o \to o'_1 \to o \to o'_2,$$
where o is the anchor edit's target object and o'_1, o'_2 are alternative target objects for the same subject–relation pattern. This produces repeated candidate updates to the same factual association. The setting stresses contradiction, target flipping, and repeated overwriting of the same local memory.
- **Easy-to-hard phase shift.** This setting tests whether a monitor remains reliable when the stream changes difficulty over time. We assign each candidate edit a hardness score based on overlap with other edits and order the stream so that low-overlap edits appear first and high-overlap edits appear later. The second phase therefore contains edits with higher expected interference. This setting stresses delayed distribution shift inside the edit stream, where early behavior may make a monitor appear calibrated before the stream becomes harder.
- **Sparse oracle.** This setting tests the same hard stream under limited full-probe evaluation. We reuse the phase-shift edit stream but change the monitoring configuration to a fixed sampling probability $q_t = 0.25$ for all t. This setting tests whether the controller remains useful when full-probe evaluations are infrequent.

D.4 Monitoring Baselines

We use the same ordered edit requests, dataset-provided prompts, probe-construction pipeline, and base-editor configuration across methods within each setting. Each method maintains its own committed model.

- **Unconstrained editing.** The unconstrained baseline commits every candidate update produced by the base editor:
$$W_t = \hat{W}_t, \qquad \mathcal{C} = \{1, \dots, T\}.$$
It uses no probe-defined risk monitor or rejection rule and measures the behavior of the underlying editor when the full submitted edit stream is applied without pre-commit control.
- **Static risk-control calibration.** This baseline adapts the risk-controlling prediction-set paradigm Angelopoulos et al. (2024b) to sequential editing. It reserves a calibration prefix $\mathcal{C}_{\mathrm{cal}} \subset \{1, \dots, T\}$ and estimates the average joint risk for each boundary $\beta \in \mathcal{G}$:
$$\widehat{D}_{\mathrm{cal}}(\beta) = \frac{1}{|\mathcal{C}_{\mathrm{cal}}|} \sum_{t \in \mathcal{C}_{\mathrm{cal}}} D_t(\beta).$$
It selects
$$\beta_{\mathrm{static}} = \inf \left\{ \beta \in \mathcal{G} : \widehat{D}_{\mathrm{cal}}(\beta) \le \theta \right\}$$
and keeps this boundary fixed for the remainder of the stream.
- **Online conformal calibration.** This baseline adapts online conformal prediction to the pre-commit decision problem Angelopoulos et al. (2024a). For each boundary $\beta \in \mathcal{G}$, it computes

the empirical $(1-\alpha)$ quantile of previously observed risks in a rolling window and declares the boundary feasible when this quantile does not exceed θ. It selects

$$\beta_t^{\mathrm{OC}} = \inf\left\{\beta \in \mathcal{G} : \widehat{Q}_t^{(1-\alpha)}(\beta) \le \theta\right\},$$

where $\widehat{Q}_t^{(1-\alpha)}(\beta)$ denotes the corresponding empirical quantile.

- **Fixed-β anytime-valid monitor.** This baseline uses sequential evidence accumulation without adaptive boundary search. Following anytime-valid inference and safe testing Ramdas et al. (2023), it fixes a boundary β_0 in advance and updates

$$M_t(\beta_0) = \prod_{i=1}^{t}\left(1 + \lambda_i\big(\widetilde{R}_i(\beta_0) - \theta\big)\right).$$

  The stream is halted when $M_t(\beta_0) \ge 1/\alpha$.
- **SAVER-Full.** This variant uses the same admissible-boundary search and martingale monitor as SAVER, but evaluates the complete online probe bank for every candidate:

$$\widetilde{R}_t(\beta) = D_t(\beta), \qquad q_t = 1, \quad K_t = 1 \quad \text{for all } t.$$

- **SAVER-Sampled.** This variant replaces full evaluation with adaptive Bernoulli sampling. The sampling probability is computed from the proxy risk estimate and bounded below by $q_{\min}$:

$$K_t \sim \mathrm{Bernoulli}(q_t), \qquad q_t = \min\left(1, \max\left(q_{\min}, \frac{\hat{r}_t}{\theta}\right)\right).$$

  Here, $\hat{r}_t$ is derived from semantic interference and temporal drift. The sampled risk estimator is

$$\widetilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\left(D_t(\beta) - \hat{r}_t\right).$$

  The selected boundary is

$$\hat{\beta}_t = \inf\left\{\beta \in \mathcal{G} : M_t(\beta) < \frac{m}{\alpha}\right\},$$

  where $m = |\mathcal{G}|$. The candidate is committed when the feasible set is non-empty and rejected otherwise.
- **Current-probe-loss gate.** This baseline evaluates the candidate's weighted generality and locality miscoverage using the current edit's probes together with probes from up to four most recently committed edits. Each probe retains its own reference target. The score uses the joint-risk weighting in equation 5 at a fixed boundary $\beta_0 = 0.99$. Lower scores receive preference in the matched-acceptance comparison. The gate uses the candidate's current probe loss without a martingale monitor or adaptive boundary search.
- **KL-drift gate.** This baseline measures the change in next-token distributions between the last committed model W_{t-1} and the candidate $\hat{W}_t$. Let $\mathcal{P}_t$ denote the concatenated collection of generality and locality prompts from the current edit and up to four most recently committed edits. Its score is

$$g_t^{\mathrm{KL}} = \frac{1}{|\mathcal{P}_t|}\sum_{x \in \mathcal{P}_t} D_{\mathrm{KL}}\big(P_{W_{t-1}}(\cdot \mid x) \,\|\, P_{\hat{W}_t}(\cdot \mid x)\big).$$

  The divergence is computed over the full next-token vocabulary. Lower drift scores receive preference in the matched-acceptance comparison.
- **Random rejection.** Let π denote the acceptance rate from a completed SAVER reference run on the same edit stream. This baseline independently accepts each request with probability π and applies the base editor only to accepted requests. Rejected requests leave the committed model unchanged. Admission is independent of probe loss and model drift, matching SAVER's acceptance rate in expectation.
- **NoProxy sampler.** This variant matches the expected full-evaluation budget of sampled SAVER, but removes semantic-interference and temporal-drift signals from the sampling rule. It uses

$$q^{\mathrm{NP}} = \frac{1}{T}\sum_{t=1}^{T} q_t^{\mathrm{SAVER}}, \qquad K_t^{\mathrm{NP}} \sim \mathrm{Bernoulli}(q^{\mathrm{NP}}).$$

  Its risk estimator is

$$\widetilde{R}_t^{\mathrm{NP}}(\beta) = \hat{r}_0 + \frac{K_t^{\mathrm{NP}}}{q^{\mathrm{NP}}}\left(D_t(\beta) - \hat{r}_0\right),$$

  where $\hat{r}_0$ is a non-adaptive baseline risk prior.

Acceptance matching. The current-probe-loss and KL-drift gates use the acceptance rate π from the completed SAVER reference run as their target, favoring candidates with lower observed scores. Random rejection uses π as its acceptance probability. We report the realized acceptance rates for all three baselines.

D.5 Evaluation Metrics

We report metrics along three axes: edit utility, preservation of chosen reference behavior, and monitoring behavior. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the submitted edit indices and $\mathcal{C} \subseteq \mathcal{T}$ denote the indices of updates committed by a method. For each edit $e_t = (s_t, r_t, o_t)$, let x_t^e be the edit prompt, $\mathcal{X}_t^{\text{in}}$ be the in-scope generality probe set, $\mathcal{X}_t^{\text{loc}}$ be the locality probe set, and $y_{\text{ref}}(x)$ denote the chosen reference for locality probe x.

- **Edit Success Rate (ESR).** ESR measures whether the committed model predicts the target object on the submitted edit prompt:

$$\text{ESR} = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} \mathbb{I}\left[\arg\max_y P_{W_t}(y \mid x_t^e) = o_t\right].$$

  Higher ESR indicates stronger direct edit utility.

- **Paraphrase Success Rate (PSR).** PSR measures whether the edit generalizes to in-scope paraphrase or equivalent prompts:

$$\text{PSR} = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} \frac{1}{|\mathcal{X}_t^{\text{in}}|} \sum_{x \in \mathcal{X}_t^{\text{in}}} \mathbb{I}\left[\arg\max_y P_{W_t}(y \mid x) = o_t\right].$$

  Higher PSR indicates better in-scope generalization of the edit. In the matched-acceptance comparison, all-request ESR and PSR use T in place of $|\mathcal{C}|$ as the denominator, with rejected requests counted as failures.

- **Neighborhood Success Rate (NSR).** NSR measures preservation of chosen reference behavior on locality probes:

$$\text{NSR} = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} \frac{1}{|\mathcal{X}_t^{\text{loc}}|} \sum_{x \in \mathcal{X}_t^{\text{loc}}} \mathbb{I}\left[\arg\max_y P_{W_t}(y \mid x) = y_{\text{ref}}(x)\right].$$

  Higher NSR indicates stronger preservation of the selected locality behavior.

- **Perplexity (PPL).** PPL is measured on the held-out reference corpus `data/ppl_reference_mini.txt`. For token sequence $(z_1, \ldots, z_n)$, we compute

$$\text{PPL} = \exp\left(-\frac{1}{n} \sum_{i=1}^{n} \log P_W(z_i \mid z_{<i})\right).$$

  Lower PPL indicates less degradation of general language-model behavior after sequential editing.

- **Joint integrity–utility risk.** For each candidate update, we compute the full-probe risk $D_t(\beta)$ defined in Equation (5). For candidate boundary β, $C(x, \beta)$ denotes the prediction set constructed using the candidate model with weights $\hat{W}_t$. The miscoverage loss is

$$L(y, C(x, \beta)) = \mathbb{I}[y \notin C(x, \beta)].$$

  The joint integrity–utility risk is

$$D_t(\beta) = a \cdot \frac{1}{|\mathcal{X}_t^{\text{loc}}|} \sum_{x \in \mathcal{X}_t^{\text{loc}}} L(y_{\text{base}}(x), C(x, \beta)) + (1 - a) \cdot \frac{1}{|\mathcal{X}_t^{\text{in}}|} \sum_{x \in \mathcal{X}_t^{\text{in}}} L(o_t, C(x, \beta)),$$

  where $y_{\text{base}}(x)$ is the reference token predicted by the initial model W_0. In the main experiments, the trade-off weight is set to $a = 0.7$. For reporting over a stream at boundary β, we average over committed updates:

$$\widehat{D}(\beta) = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} D_t(\beta).$$

  Lower $\widehat{D}$ indicates lower joint integrity–utility risk.

- **Selected admissible boundary.** $\hat{\beta}_t$ denotes the boundary returned by the monitor at step t. For SAVER, this is the smallest admissible boundary found on the candidate grid:

$$\hat{\beta}_t = \inf\left\{\beta \in \mathcal{G} : M_t(\beta) < \frac{m}{\alpha}\right\},$$

whenever the feasible set is non-empty, where $m = |\mathcal{G}|$. For fixed-β baselines, $\hat{\beta}_t$ is the fixed operating boundary.

- **Acceptance rate (Acc.).** Acceptance rate is the fraction of submitted candidate updates committed by the method:

$$\text{Acc.} = \frac{|\mathcal{C}|}{T}.$$

Higher Acc. means the monitor admits more of the submitted stream.

- **Sample Rate.** Sample Rate denotes the realized fraction of candidate updates receiving full-probe evaluation:

$$\text{Sample Rate} = \frac{1}{T}\sum_{t=1}^{T} K_t.$$

Lower Sample Rate indicates fewer full-probe evaluations.

- **Wall-clock time (Time).** Time is the total wall-clock runtime, in hours, required to process the T submitted candidate updates under the corresponding monitoring policy.
- **Empirical risk-estimate variance (Var.).** For sampled monitoring, the per-step estimator is

$$\widetilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\left(D_t(\beta) - \hat{r}_t\right).$$

Let $\widetilde{R}_t^{\log}$ denote the risk estimate recorded at round t for this statistic. We report its empirical variance over the stream:

$$\text{Var.} = \frac{1}{T-1}\sum_{t=1}^{T}\left(\widetilde{R}_t^{\log} - \frac{1}{T}\sum_{s=1}^{T}\widetilde{R}_s^{\log}\right)^2.$$

Lower Var. indicates less variation in the recorded risk estimates.

E Additional results

E.1 Results w.r.t. Long-Run Editing with Existing Editors (Cont.)

E.1.1 Results w.r.t different LLM Backbone

Table 5 provides the corresponding RQ1 results on Llama-3.1-8B. The trends are consistent with the Qwen2.5-7B results in the main paper. Wrapping existing editors with SAVER improves NSR and reduces PPL across all editor–dataset pairs, while keeping Acc. high for stronger editors such as AlphaEdit, UltraEdit, and WISE. The largest relative gains again appear for the more distortion-prone editors, such as ROME and MEMIT, where SAVER substantially improves NSR and reduces PPL after 500 submitted candidate updates. These results support the claim that SAVER acts as a model- and editor-agnostic pre-commit control layer while the underlying editor still determines how candidate updates are written, but SAVER consistently improves the quality of the committed stream across different backbone LLMs.

Table 5: Long-run editing with existing editors on **CounterFact** and **zsRE** w and w/o SAVER for **Llama-3.1-8B** in terms of Acc., ESR, PSR, NSR, and PPL over 500 submitted candidate updates. Parentheses show SAVER's relative change over the unconstrained editor. Results are averaged over 5 seeds, $p < 0.05$ by a paired two-sided test vs the best competing method.

Editor	Method	CounterFact					zsRE				
		Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓
ROME	Unconstr.	1.000	0.188	0.112	0.030	98.48	1.000	0.164	0.185	0.020	86.82
	+SAVER	0.841	0.228 (+21.3%)	0.123 (+9.8%)	0.060 (+100.0%)	88.85 (-9.8%)	0.873	0.198 (+20.7%)	0.186 (+0.5%)	0.029 (+45.0%)	78.78 (-9.3%)
MEMIT	Unconstr.	1.000	0.210	0.127	0.047	92.50	1.000	0.191	0.195	0.025	83.18
	+SAVER	0.886	0.214 (+1.9%)	0.155 (+22.0%)	0.066 (+40.4%)	84.32 (-8.8%)	0.895	0.238 (+24.6%)	0.214 (+9.7%)	0.055 (+120.0%)	74.94 (-9.9%)
WISE	Unconstr.	1.000	0.245	0.209	0.076	59.57	1.000	0.180	0.204	0.066	62.68
	+SAVER	0.989	0.298 (+21.6%)	0.225 (+7.7%)	0.101 (+32.9%)	49.62 (-16.7%)	0.975	0.245 (+36.1%)	0.229 (+12.3%)	0.094 (+42.4%)	58.00 (-7.5%)
UltraEdit	Unconstr.	1.000	0.354	0.311	0.058	45.49	1.000	0.392	0.263	0.048	46.03
	+SAVER	0.972	0.432 (+22.0%)	0.356 (+14.5%)	0.085 (+46.6%)	42.12 (-7.4%)	0.998	0.408 (+4.1%)	0.291 (+10.6%)	0.063 (+31.3%)	39.38 (-14.4%)
AlphaEdit	Unconstr.	1.000	0.882	0.599	0.132	48.20	1.000	0.702	0.551	0.032	50.83
	+SAVER	0.996	0.904 (+2.5%)	0.619 (+3.3%)	0.163 (+23.5%)	40.46 (-16.1%)	0.965	0.774 (+10.3%)	0.554 (+0.5%)	0.037 (+15.6%)	43.94 (-13.6%)

E.1.2 MATCHED-ACCEPTANCE COMPARISON

We compare SAVER-Sampled with current-probe-loss, KL-drift, and random-rejection gates on CounterFact using Qwen2.5-7B + AlphaEdit over $T = 500$ submitted edits. The baseline implementations are described in Appendix D.4.

Table 6: Comparison with admission gates at comparable acceptance rates on **CounterFact** using **Qwen2.5-7B + AlphaEdit**. All-request ESR and PSR count rejected requests as failures.

Method	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	$\widehat{D}\downarrow$
SAVER-Sampled	0.986	**0.866**	**0.618**	**0.157**	**41.67**	**0.095**
Current-probe-loss gate	0.985	0.847	0.599	0.142	45.36	0.116
KL-drift gate	0.986	0.841	0.592	0.148	44.52	0.110
Random rejection	0.986	0.831	0.582	0.127	49.12	0.142

As shown in Table 6, acceptance rates lie between 0.985 and 0.987, while SAVER achieves higher ESR, PSR, and NSR and lower PPL and joint risk than all three gates. Its all-request ESR and PSR reach 0.866 and 0.618, compared with 0.847 and 0.599 for the current-probe-loss gate. The KL-drift gate provides the strongest preservation among the competing gates, but SAVER further improves NSR from 0.148 to 0.157 and reduces PPL from 44.52 to 41.67. Random rejection yields the lowest NSR and highest PPL among the compared methods. These results support the effectiveness of SAVER's admission decisions at comparable acceptance rates.

E.1.3 LONGER EDIT STREAMS

We extend the CounterFact evaluation to $T \in \{100, 500, 1000, 2500, 5000\}$ submitted edits using Qwen3.5-9B + AlphaEdit. At each stream length, we compare unconstrained editing with its SAVER-Sampled variant.

Table 7: Sequential editing on **CounterFact** using **Qwen3.5-9B + AlphaEdit** at increasing stream lengths. T denotes the number of submitted edits.

T	Method	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓	$\widehat{D}\downarrow$	$\hat{\beta}_T$
100	Unconstrained	1.000	0.886	0.640	0.190	34.70	0.108	–
	+SAVER	0.992	**0.901**	**0.653**	**0.205**	**33.50**	**0.087**	0.820
500	Unconstrained	1.000	0.851	0.602	0.133	47.90	0.144	–
	+SAVER	0.987	**0.872**	**0.624**	**0.163**	**40.80**	**0.094**	0.840
1,000	Unconstrained	1.000	0.801	0.552	0.104	64.30	0.181	–
	+SAVER	0.979	**0.848**	**0.598**	**0.151**	**47.20**	**0.093**	0.835
2,500	Unconstrained	1.000	0.695	0.455	0.069	103.50	0.271	–
	+SAVER	0.967	**0.792**	**0.544**	**0.124**	**59.80**	**0.098**	0.860
5,000	Unconstrained	1.000	0.574	0.357	0.044	162.70	0.348	–
	+SAVER	0.952	**0.731**	**0.481**	**0.102**	**76.40**	**0.097**	0.855

Table 7 shows that SAVER achieves higher ESR, PSR, and NSR and lower PPL and joint risk at every reported stream length. The performance gap widens as more edits are submitted. At $T = 5{,}000$, SAVER commits 95.2% of requests while improving ESR from 0.574 to 0.731 and NSR from 0.044 to 0.102. PPL decreases from 162.70 to 76.40, a reduction of 53.0%.

Across the reported lengths, SAVER's average joint risk remains between 0.087 and 0.098, below the default threshold $\theta = 0.10$, while unconstrained risk increases from 0.108 to 0.348. Although editing performance declines as the stream grows, SAVER limits the observed degradation while continuing to admit most requested updates.

E.2 ADDITIONAL RESULTS ON USER-SPECIFIED RISK CONTROL

E.2.1 DETECTION UNDER PERSISTENT EXCESS RISK

We initialize the monitor with 100 below-threshold rounds and then introduce persistent excess risk. We use $\theta = 0.10$, $\alpha = 0.10$, and full-probe evaluation with $q_t = 1$. We examine the fixed boundary $\beta = 0.99$ and the 23-point grid $\mathcal{G} = \{0.55, 0.57, \ldots, 0.99\}$. Results are summarized over 2,000 replications with a 2,000-round post-change horizon. Detection delay is measured from the change point to threshold crossing.

As shown in Table 8, larger persistent excess risk leads to shorter detection delays. As excess risk increases from 0.037 to 0.233, the median fixed-boundary delay decreases from 400 to 64 rounds,

Table 8: Detection under controlled persistent excess risk. Delays are reported in monitoring rounds; P90 denotes the 90th percentile. The final column reports the percentage detected within the post-change horizon.

Post-change risk	Excess risk	Fixed-β delay Median / P90	Grid delay Median / P90	Pre-detection excess-risk commits Median	Detected within 2,000 rounds
0.137	0.037	400 / 560	535 / 715	534	$\geq 99.5\%$
0.236	0.136	100 / 122	123 / 147	122	100%
0.333	0.233	64 / 75	78 / 89	77	100%

while the median grid delay decreases from 535 to 78 rounds. The 90th-percentile delays also decrease, showing that stronger sustained violations produce earlier detection across replications.

E.3 Results w.r.t. Robustness under Hard Edit Settings

We further test whether the pre-commit controller remains reliable under edit streams designed to amplify sequential interference. We use four controlled edit-stream settings, defined in Appendix D.3: **(1)** overlap bursts, **(2)** contradictory re-edit loops, **(3)** easy-to-hard phase shifts, and **(4)** and sparse oracle evaluation These settings isolate failure modes that can be hidden in randomly ordered edit streams. We evaluate unconstrained AlphaEdit and SAVER-wrapped AlphaEdit under the same Qwen2.5-7B setting. As shown in Table 9, SAVER improves ESR, PSR, NSR, and PPL across all four controlled edit-stream settings while committing fewer updates. The pattern is most pronounced when the stream creates direct overwrite pressure: in the contradictory re-edit loop, unconstrained editing has the weakest utility and preservation scores, and SAVER improves ESR, PSR, and NSR by 51.0%, 59.5%, and 51.3%, respectively. This suggests that repeated target flips are especially harmful when every candidate update is committed without a pre-commit admissibility check. The phase-shift setting shows a complementary failure mode: after an initially easier prefix, high-overlap edits later in the stream increase reference-behavior damage, and SAVER improves NSR from 0.095 to 0.135 while reducing PPL from 48.21 to 42.04. Under sparse oracle evaluation, SAVER still improves ESR, PSR, and NSR while reducing PPL by 12.6%, indicating that interference-aware monitoring remains useful when full-probe evaluation is limited. Overall, RQ4 shows that SAVER's selectivity increases in harder streams, yielding a committed stream with lower post-edit distortion rather than merely fewer updates.

Table 9: Robustness analysis under controlled hard edit-stream settings on **CounterFact**. Parentheses show SAVER's relative change over the unconstrained editor. Results are averaged over 5 seeds, $p < 0.05$ by a paired two-sided test vs the best competing method.

Setting	Method	Acc.↑	ESR↑	PSR↑	NSR↑	PPL↓
Overlap	Unconstr.	1.000	0.724	0.499	0.115	47.49
	+SAVER	0.829	0.768 (+6.1%)	0.525 (+5.2%)	0.122 (+6.1%)	40.01 (-15.8%)
Contradict.	Unconstr.	1.000	0.247	0.153	0.039	44.78
	+SAVER	0.736	0.373 (+51.0%)	0.244 (+59.5%)	0.059 (+51.3%)	41.27 (-7.8%)
Phase	Unconstr.	1.000	0.750	0.494	0.095	48.21
	+SAVER	0.792	0.787 (+4.9%)	0.573 (+16.0%)	0.135 (+42.1%)	42.04 (-12.8%)
Sparse	Unconstr.	1.000	0.750	0.494	0.095	48.51
	+SAVER	0.814	0.799 (+6.5%)	0.522 (+5.7%)	0.120 (+26.3%)	42.42 (-12.6%)

E.4 Sensitivity to Probe Coverage and Quality

We evaluate probe sensitivity on CounterFact using Qwen2.5-7B + AlphaEdit, following the construction in Appendix D.2. Here, $\widehat{D}_{\mathrm{sel}}$ denotes the joint risk on the monitoring probes, while $\widehat{D}_{\mathrm{hold}}$ and $\mathrm{NSR}_{\mathrm{hold}}$ measure locality miscoverage and reference preservation on the disjoint audit probes. We report the Spearman correlation ρ_D between monitored and held-out risks on rounds where both are evaluated.

Probe coverage. We vary the fraction of monitoring locality probes from 25% to 100%, keeping the generality probes and held-out audit set fixed.

Table 10: Monitoring locality-probe coverage on **CounterFact** using **Qwen2.5-7B + AlphaEdit** over $T = 500$ submitted edits. Generality probes and held-out locality audit probes remain fixed.

Locality-probe coverage	Acc.↑	$\widehat{D}_{\mathrm{sel}}\downarrow$	$\widehat{D}_{\mathrm{hold}}\downarrow$	$\mathrm{NSR}_{\mathrm{hold}}\uparrow$	$\rho_D\uparrow$	$\hat{\beta}_T$
25%	0.977	0.088	0.121	0.142	0.421	0.880
50%	0.982	0.093	0.109	0.150	0.548	0.850
75%	0.981	0.091	0.103	0.152	0.603	0.860
100%	0.986	0.094	0.095	0.157	0.667	0.835

As shown in Table 10, increasing coverage reduces held-out locality miscoverage from 0.121 to 0.095 and improves held-out NSR from 0.142 to 0.157. The corresponding risk correlation increases from 0.421 to 0.667, indicating stronger agreement between monitored and held-out behavior.

Probe quality. We replace the monitoring locality prompts with semantically unrelated donor prompts of the same set size. Generality probes and the held-out locality audit set remain fixed.

Table 11: Monitoring locality-probe quality on **CounterFact** using **Qwen2.5-7B + AlphaEdit**. Standard locality prompts are compared with semantically unrelated donor prompts, keeping generality probes and held-out locality audit probes fixed.

T	**Locality probes**	**Acc.**↑	$\widehat{D}_{\mathrm{sel}}$ ↓	$\widehat{D}_{\mathrm{hold}}$ ↓	$\mathrm{NSR}_{\mathrm{hold}}$ ↑	ρ_D ↑	$\hat{\beta}_T$
100	Standard	0.989	0.096	0.101	0.179	0.592	0.825
	Semantically unrelated	0.991	0.089	0.116	0.163	0.322	0.775
250	Standard	0.987	0.093	0.097	0.166	0.641	0.840
	Semantically unrelated	0.990	0.090	0.119	0.146	0.281	0.785
500	Standard	0.986	0.094	0.095	0.157	0.667	0.835
	Semantically unrelated	0.991	0.086	0.124	0.140	0.274	0.780

In Table 11, unrelated monitoring prompts yield lower selected-set risk but higher held-out locality miscoverage at every stream length. At $T = 500$, held-out miscoverage increases from 0.095 to 0.124, while NSR decreases from 0.157 to 0.140 and risk correlation falls from 0.667 to 0.274. These results show the importance of monitoring probes that represent the locality behavior being preserved.

E.5 Results w.r.t. Component Ablations

We finally isolate three control decisions in SAVER: **(1)** where to spend full-probe evaluations, **(2)** which admissible boundary to use, and **(3)** whether sequential evidence should accumulate across the stream. SAVER-Full evaluates every candidate update, while SAVER-Sampled is the practical version used in the main experiments. As shown in Table 12, the sampled variant closely preserves the behavior of full evaluation: on CounterFact, it matches SAVER-Full on NSR, slightly improves PSR and PPL, and differs by only 0.001 in Acc.; on zsRE, Full has slightly better PSR/NSR/PPL, while Sampled has higher Acc. and ESR. Removing adaptive sampling weakens preservation and fluency, with PPL rising from 41.67 to 44.65 on CounterFact and from 42.61 to 46.59 on zsRE. Removing adaptive boundary search makes the controller rigid: Fixed β commits far fewer updates, especially on zsRE where Acc. falls to 0.768, but it still does not match the preservation quality of SAVER-Sampled. Removing the martingale weakens the committed stream even when acceptance is similar: on CounterFact, No martingale has nearly the same Acc. as SAVER-Sampled, but PPL worsens from 41.67 to 44.56 and NSR drops from 0.157 to 0.145. Overall, the ablations show that SAVER's components address distinct failure modes: adaptive sampling allocates monitoring budget, adaptive boundary search avoids a stale operating point, and the martingale monitor accumulates sequential evidence across the stream.

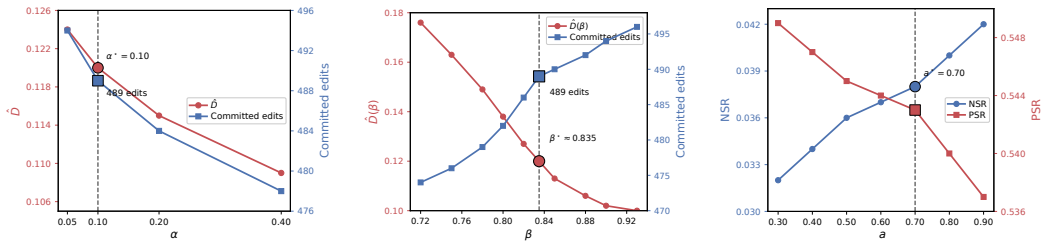
Table 12: Component ablations on **CounterFact** and **zsRE**. Best controlled values are in **bold**; second best are underlined.

Dataset	**Method**	**Acc.**↑	**ESR**↑	**PSR**↑	**NSR**↑	**PPL**↓
CounterFact	SAVER-Full	**0.987**	**0.886**	0.625	**0.157**	41.78
	SAVER-Sampled	0.986	0.878	**0.627**	**0.157**	**41.67**
	Fixed sample	0.922	0.845	0.590	0.131	44.65
	Fixed β	0.806	0.844	0.601	0.139	46.18
	No mart.	0.986	0.851	0.603	0.145	44.56
	Unconstr.	1.000	0.844	0.590	0.126	49.64
zsRE	SAVER-Full	0.967	0.738	**0.548**	**0.041**	**42.20**
	SAVER-Sampled	0.978	**0.746**	0.543	0.038	42.61
	Fixed sample	0.964	0.731	0.529	0.034	46.59
	Fixed β	0.768	0.735	0.531	0.034	45.51
	No mart.	**0.979**	0.735	0.533	0.034	45.51
	Unconstr.	1.000	0.724	0.526	0.033	48.86

E.6 Parameter Sensitivity Analysis

We further analyze how the main control parameters affect the behavior of SAVER.We conduct experiments using AlphaEdit by Qwen2.5-7B on zsRE. Specifically, we study the sequential significance level α, the admissible boundary β, and the utility–preservation trade-off weight a. This analysis is used to understand how implementation-level control choices affect conservatism, accepted edits and preservation behavior.

Effect of the sequential significance level α. We first vary α while keeping the risk threshold and boundary grid fixed. As shown in Figure 6(a), the achieved post-hoc risk $\widehat{D}$ changes only mildly across the tested values, while the number of committed edits remains close to the selected operating point. We use $\alpha^\star = 0.10$ in the main experiments because it provides a stable balance between risk



(a) Varying α. (b) Varying β. (c) Varying a.

Figure 6: Parameter analysis for SAVER. (a) Effect of the sequential significance level α on achieved risk $\widehat{D}$ and committed edits. (b) Effect of the admissible boundary β on achieved risk $\widehat{D}(\beta)$ and committed edits; the selected operating point is $\beta^\star \approx 0.835$. (c) Effect of the joint-risk weight a on NSR and PSR; larger a places more emphasis on preservation of chosen reference behavior.

monitoring and edit capacity. This sweep indicates that the empirical behavior of SAVER is not overly sensitive to small changes around the default significance level.

Effect of the admissible boundary β. We next vary β to evaluate how the admissible boundary affects probe-defined risk and edit admission. Increasing β expands the prediction sets, reducing reference-target miscoverage for a fixed candidate. As shown in Figure 6(b), the reported risk $\widehat{D}(\beta)$ decreases while the number of committed edits increase across the tested values. The selected operating point $\beta^\star \approx 0.835$ yields 489 committed edits.

Effect of the utility–preservation weight a. Finally, we vary the trade-off weight a in the joint post-edit distortion risk. Larger a places more emphasis on locality/reference preservation, while smaller a emphasizes in-scope edit generalization. As shown in Figure 6(c), increasing a improves NSR but slightly reduces PSR, matching the intended interpretation of the weighted joint risk. We select $a^\star = 0.70$ as the default setting because it improves reference-behavior preservation while retaining competitive paraphrase generalization. Overall, the parameter analysis confirms that the main control parameters have interpretable effects and provides guidance for selecting operating points in different editing settings.

F Proofs

F.1 Proof of Lemma 1

Proof. We begin from the definition of the importance-sampled estimator:

$$\tilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big). \tag{i}$$

Taking the conditional expectation on both sides with respect to $\mathcal{F}_{t-1}$ and $D_t(\beta)$ yields:

$$\begin{aligned}\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big] &= \mathbb{E}\left[\hat{r}_t + \frac{K_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big) \;\middle|\; \mathcal{F}_{t-1}, D_t(\beta)\right] \\ &= \hat{r}_t + \mathbb{E}\left[\frac{K_t}{q_t} \;\middle|\; \mathcal{F}_{t-1}, D_t(\beta)\right]\big(D_t(\beta) - \hat{r}_t\big),\end{aligned} \tag{ii}$$

where $\hat{r}_t$ and q_t are treated as constants due to their $\mathcal{F}_{t-1}$-measurability.

Since $K_t \mid \mathcal{F}_{t-1} \sim \text{Bernoulli}(q_t)$ and K_t is conditionally independent of $D_t(\beta)$ given $\mathcal{F}_{t-1}$, it follows that:

$$\mathbb{E}[K_t \mid \mathcal{F}_{t-1}, D_t(\beta)] = \mathbb{E}[K_t \mid \mathcal{F}_{t-1}] = q_t. \tag{iii}$$

Therefore, the expected sampling weight is:

$$\mathbb{E}\left[\frac{K_t}{q_t} \;\middle|\; \mathcal{F}_{t-1}, D_t(\beta)\right] = \frac{1}{q_t}\,\mathbb{E}[K_t \mid \mathcal{F}_{t-1}, D_t(\beta)] = 1. \tag{iv}$$

Substituting Eq. iv into Eq. ii, we obtain:

$$\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big] = \hat{r}_t + \big(D_t(\beta) - \hat{r}_t\big) = D_t(\beta). \tag{v}$$

Next, we apply the tower property of conditional expectation to compute the expectation given only the history $\mathcal{F}_{t-1}$:

$$\begin{aligned}\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big] \;\Big|\; \mathcal{F}_{t-1}\Big] \\ &= \mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}]\,. \end{aligned} \tag{vi}$$

Consequently, under the null condition $\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \le \theta$, it follows immediately that:

$$\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big] \le \theta. \tag{vii}$$

Hence Proved. □

F.2 Proof of Theorem 1

Proof. We first compute the conditional variance of the estimator given both the history $\mathcal{F}_{t-1}$ and the true risk $D_t(\beta)$. We begin from the definition of the importance-sampled estimator:

$$\tilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big). \tag{i}$$

For simplicity, let's define the prediction error as:

$$\Delta_t(\beta) = D_t(\beta) - \hat{r}_t. \tag{ii}$$

Then Eq. i becomes,

$$\tilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\Delta_t(\beta). \tag{iii}$$

Since $\hat{r}_t$ is $\mathcal{F}_{t-1}$-measurable, conditioning on $\mathcal{F}_{t-1}$ and $D_t(\beta)$ treats both $\hat{r}_t$ and $D_t(\beta)$ as fixed constants. Therefore,

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big) = \mathrm{Var}\left(\frac{K_t}{q_t}\Delta_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\right). \tag{iv}$$

When we factor out the constant $\Delta_t(\beta)$, we get:

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big) = \Delta_t(\beta)^2\,\mathrm{Var}\left(\frac{K_t}{q_t} \mid \mathcal{F}_{t-1}, D_t(\beta)\right). \tag{v}$$

By assumption, $K_t \mid \mathcal{F}_{t-1} \sim \mathrm{Bernoulli}(q_t)$, and K_t is conditionally independent of $D_t(\beta)$ given $\mathcal{F}_{t-1}$. Therefore we can write:

$$\mathbb{E}\left[\frac{K_t}{q_t} \mid \mathcal{F}_{t-1}, D_t(\beta)\right] = \frac{1}{q_t}\mathbb{E}[K_t \mid \mathcal{F}_{t-1}] = 1, \tag{vi}$$

and

$$\mathbb{E}\left[\left(\frac{K_t}{q_t}\right)^2 \middle| \mathcal{F}_{t-1}, D_t(\beta)\right] = \frac{1}{q_t^2}\mathbb{E}[K_t \mid \mathcal{F}_{t-1}] = \frac{1}{q_t}. \tag{vii}$$

Hence, the variance can be denoted as:

$$\mathrm{Var}\left(\frac{K_t}{q_t} \mid \mathcal{F}_{t-1}, D_t(\beta)\right) = \frac{1}{q_t} - 1 = \frac{1-q_t}{q_t}. \tag{viii}$$

Substituting Eq. viii into Eq. v gives us the exact conditional variance:

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big) = \frac{1-q_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big)^2. \tag{ix}$$

We now use this identity and the law of total variance to evaluate the total conditional variance given only $\mathcal{F}_{t-1}$:

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big) = \mathbb{E}\Big[\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big) \;\Big|\; \mathcal{F}_{t-1}\Big] + \mathrm{Var}\Big(\mathbb{E}\Big[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)\Big] \;\Big|\; \mathcal{F}_{t-1}\Big). \tag{x}$$

By Lemma 1, $\mathbb{E}[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}, D_t(\beta)] = D_t(\beta)$. Substituting this and Eq. ix into Eq. x, and pulling the $\mathcal{F}_{t-1}$-measurable constant $(1-q_t)/q_t$ out of the inner expectation, we obtain:

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big) = \frac{1-q_t}{q_t}\mathbb{E}\Big[\big(D_t(\beta) - \hat{r}_t\big)^2 \;\Big|\; \mathcal{F}_{t-1}\Big] + \mathrm{Var}(D_t(\beta) \mid \mathcal{F}_{t-1})\,. \tag{xi}$$

Now consider the standard inverse propensity estimator (IPW) without a control variate, whose variance is given by $\mathrm{Var}(R_t^{IPW}(\beta) \mid \mathcal{F}_{t-1})$. Repeating the exact same conditioning steps for this naive estimator, its inner conditional variance is $\frac{1-q_t}{q_t} D_t(\beta)^2$. Applying the law of total variance again gives:

$$\mathrm{Var}\Big(R_t^{IPW}(\beta) \mid \mathcal{F}_{t-1}\Big) = \frac{1-q_t}{q_t}\mathbb{E}\big[D_t(\beta)^2 \;\big|\; \mathcal{F}_{t-1}\big] + \mathrm{Var}(D_t(\beta) \mid \mathcal{F}_{t-1})\,. \tag{xii}$$

Under the stated theorem condition, we know $\mathbb{E}\Big[\big(D_t(\beta) - \hat{r}_t\big)^2 \mid \mathcal{F}_{t-1}\Big] \le \mathbb{E}\big[D_t(\beta)^2 \mid \mathcal{F}_{t-1}\big]$, and since $\frac{1-q_t}{q_t} \ge 0$, comparing Eq. xi and Eq. xii gives us:

$$\mathrm{Var}\Big(\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}\Big) \le \mathrm{Var}\Big(R_t^{IPW}(\beta) \mid \mathcal{F}_{t-1}\Big)\,. \tag{xiii}$$

Hence Proved. □

F.3 Proof of Theorem 2

Proof. Let $\tau \coloneqq m/\alpha$. We begin by fixing any two coverage levels $\beta_1, \beta_2 \in [\beta_{min}, \beta_{max}]$ such that $\beta_1 \le \beta_2$. By definition, the set-valued predictor is nested, i.e.,

$$C(x, \beta_1) \subseteq C(x, \beta_2) \qquad \text{for every probe } x. \tag{i}$$

Since $C(x, \beta_1) \subseteq C(x, \beta_2)$, whenever the target token $y \in C(x, \beta_1)$ we also have $y \in C(x, \beta_2)$. Therefore, the miscoverage loss is bounded:

$$\mathbb{I}(y \notin C(x, \beta_2)) \le \mathbb{I}(y \notin C(x, \beta_1))\,. \tag{ii}$$

Because the true risk $D_t(\beta)$ is the expectation over this indicator loss, it follows immediately that,

$$D_t(\beta_2) \le D_t(\beta_1). \tag{iii}$$

We now turn to the importance-sampled estimator $\tilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big)$. By assumption, $\hat{r}_t$ and q_t do not depend on β, and $\frac{K_t}{q_t} \ge 0$. Therefore, Eq. iii implies:

$$\tilde{R}_t(\beta_2) \le \tilde{R}_t(\beta_1). \tag{iv}$$

The Edit Risk Martingale (ERM)$M_t(\beta)$ is the product:

$$M_t(\beta) = \prod_{i=1}^{t} \Big(1 + \lambda_i^*(\beta)\big(\tilde{R}_i(\beta) - \theta\big)\Big)\,. \tag{v}$$

The oracle monitor $M_t^\star(\beta)$ replaces $\tilde{R}_i(\beta)$ with $D_i(\beta)$ along the same candidate sequence and using the same sensitivities. By the theorem assumption, $\overline{\mathcal{B}}_t$ and $\mathcal{B}_t^\star$ are non-empty right-sided intervals, and both monitors are strictly positive. Hence, the infima $\bar{\beta}_t = \inf \overline{\mathcal{B}}_t$ and $\beta_t^\star = \inf \mathcal{B}_t^\star$ are well-defined.

Next, we bound the distance between the empirical and oracle boundaries. Let $\Psi_t(\beta) \coloneqq \log M_t(\beta)$ and $\Psi_t^\star(\beta) \coloneqq \log M_t^\star(\beta)$. We first prove the lower bound on $\bar{\beta}_t$. Let $\eta > 0$ and define

$$\underline{\beta}_\eta \coloneqq \max\left\{\beta_{min},\, \beta_t^\star - \frac{\varepsilon_t}{m_t} - \eta\right\}. \tag{vi}$$

If $\beta_t^\star - \frac{\varepsilon_t}{m_t} \le \beta_{min}$, the bound $\bar{\beta}_t \ge \beta_t^\star - \frac{\varepsilon_t}{m_t}$ holds trivially because $\bar{\beta}_t \ge \beta_{min}$. In the remaining case, choose $0 < \eta < \min\left\{\delta_{\mathrm{loc}} - \frac{\varepsilon_t}{m_t},\, \beta_t^\star - \frac{\varepsilon_t}{m_t} - \beta_{min}\right\}$. Then $\underline{\beta}_\eta = \beta_t^\star - \frac{\varepsilon_t}{m_t} - \eta < \beta_t^\star$, and:

$$\underline{\beta}_\eta \in [\beta_t^\star - \delta_{\mathrm{loc}}, \beta_t^\star + \delta_{\mathrm{loc}}] \cap [\beta_{min}, \beta_{max}]. \tag{vii}$$

Because $\underline{\beta}_\eta < \beta_t^\star$, the sign-aware local margin assumption gives:

$$\Psi_t^\star(\underline{\beta}_\eta) - \log\tau \ge m_t(\beta_t^\star - \underline{\beta}_\eta) = \varepsilon_t + m_t\eta. \tag{viii}$$

Since $\underline{\beta}_\eta$ lies to the left of the oracle boundary, it is oracle-infeasible, yielding:

$$\Psi_t^\star(\underline{\beta}_\eta) \ge \log\tau + \varepsilon_t + m_t\eta. \tag{ix}$$

Applying the uniform log-martingale error assumption $|\Psi_t(\beta) - \Psi_t^\star(\beta)| \le \varepsilon_t$, we get

$$\Psi_t(\underline{\beta}_\eta) \ge \Psi_t^\star(\underline{\beta}_\eta) - \varepsilon_t \ge \log\tau + m_t\eta > \log\tau. \tag{x}$$

Therefore, $\underline{\beta}_\eta \notin \overline{\mathcal{B}}_t$. Since $\overline{\mathcal{B}}_t$ is a right-sided interval, this implies $\bar{\beta}_t \ge \underline{\beta}_\eta$. Letting $\eta \downarrow 0$, we obtain

$$\bar{\beta}_t \ge \beta_t^\star - \frac{\varepsilon_t}{m_t}. \tag{xi}$$

We then prove the upper bound. Let $\eta > 0$ and define,

$$\bar{\beta}_\eta \coloneqq \min\left\{\beta_{max},\, \beta_t^\star + \frac{\varepsilon_t}{m_t} + \eta\right\}. \tag{xii}$$

If $\beta_t^\star + \frac{\varepsilon_t}{m_t} \ge \beta_{max}$, the bound $\bar{\beta}_t \le \beta_t^\star + \frac{\varepsilon_t}{m_t}$ holds trivially because $\bar{\beta}_t \le \beta_{max}$. In the remaining case, choose $0 < \eta < \min\left\{\delta_{\mathrm{loc}} - \frac{\varepsilon_t}{m_t},\, \beta_{max} - \beta_t^\star - \frac{\varepsilon_t}{m_t}\right\}$. Then $\bar{\beta}_\eta = \beta_t^\star + \frac{\varepsilon_t}{m_t} + \eta > \beta_t^\star$ lies within the domain and the local radius δ_{loc}. The sign-aware local margin assumption gives us:

$$\log\tau - \Psi_t^\star(\bar{\beta}_\eta) \ge m_t(\bar{\beta}_\eta - \beta_t^\star) = \varepsilon_t + m_t\eta. \tag{xiii}$$

Since $\bar{\beta}_\eta$ lies to the right of the oracle boundary, it is oracle-feasible, hence:

$$\Psi_t^\star(\bar{\beta}_\eta) \le \log\tau - \varepsilon_t - m_t\eta. \tag{xiv}$$

Applying the uniform log-martingale error assumption, we get:

$$\Psi_t(\bar{\beta}_\eta) \le \Psi_t^\star(\bar{\beta}_\eta) + \varepsilon_t \le \log\tau - m_t\eta < \log\tau. \tag{xv}$$

Therefore, $\bar{\beta}_\eta \in \overline{\mathcal{B}}_t$. Since $\bar{\beta}_t = \inf\overline{\mathcal{B}}_t$, this implies $\bar{\beta}_t \le \bar{\beta}_\eta$. Letting $\eta \downarrow 0$, we obtain,

$$\bar{\beta}_t \le \beta_t^\star + \frac{\varepsilon_t}{m_t}. \tag{xvi}$$

Finally, combining Eq. xi and Eq. xvi, we conclude that:

$$\left|\bar{\beta}_t - \beta_t^\star\right| \le \frac{\varepsilon_t}{m_t}. \tag{xvii}$$

This proves the theorem. □

F.4 Proof of Theorem 3

Proof. We begin by writing the Edit Risk Martingale in its recursive form. Let:

$$B_t(\beta) \;\coloneqq\; 1 + \lambda_t\big(\tilde{R}_t(\beta) - \theta\big). \tag{i}$$

Then, by construction of the process, we can write:

$$M_t(\beta) = M_{t-1}(\beta)\, B_t(\beta), \qquad M_0(\beta) = 1. \tag{ii}$$

For the anytime-valid guarantee, we need to verify two key properties: a) that the process remains non-negative; and b) that it is a supermartingale.

We first show that the martingale factor in Eq. i is non-negative. We know that:

$$\tilde{R}_t(\beta) = \hat{r}_t + \frac{K_t}{q_t}\big(D_t(\beta) - \hat{r}_t\big).$$

If $K_t = 0$, then $\tilde{R}_t(\beta) = \hat{r}_t \in [0,1]$. If $K_t = 1$, then we have:

$$\tilde{R}_t(\beta) = \hat{r}_t + \frac{1}{q_t}\big(D_t(\beta) - \hat{r}_t\big) = \frac{1}{q_t}D_t(\beta) + \Big(1 - \frac{1}{q_t}\Big)\hat{r}_t. \tag{iii}$$

Since both $D_t(\beta)$ and $\hat{r}_t$ lie in $[0,1]$, the minimum possible value of Eq. iii is attained at $D_t(\beta) = 0$ and $\hat{r}_t = 1$, which gives:

$$\tilde{R}_t(\beta) \ge 1 - \frac{1}{q_t} = -\Big(\frac{1}{q_t} - 1\Big) \ge -\Big(\frac{1}{q_{\min}} - 1\Big), \tag{iv}$$

where we used the fact that $q_t \ge q_{\min}$.

Substituting Eq. iv into Eq. i, we obtain:

$$B_t(\beta) = 1 + \lambda_t\big(\tilde{R}_t(\beta) - \theta\big) \geq 1 - \lambda_t\Big(\frac{1}{q_{\min}} - 1 + \theta\Big). \tag{v}$$

Since the predictable sensitivity λ_t satisfies the admissible range $0 \leq \lambda_t \leq (\frac{1}{q_{\min}} - 1 + \theta)^{-1}$, the right-hand side of Eq. v is non-negative. Therefore,

$$B_t(\beta) \geq 0 \qquad \text{for all } t. \tag{vi}$$

Since $M_0(\beta) = 1$ and $M_t(\beta) = M_{t-1}(\beta)B_t(\beta)$, from the inductive logic it follows that:

$$M_t(\beta) \geq 0 \qquad \text{for all } t. \tag{vii}$$

We now show that $\{M_t(\beta)\}_{t\geq 1}$ is a supermartingale. Since $M_{t-1}(\beta)$, q_t, $\hat{r}_t$, and λ_t are $\mathcal{F}_{t-1}$-measurable, taking conditional expectation of Eq. ii with respect to $\mathcal{F}_{t-1}$ gives us:

$$\begin{aligned}\mathbb{E}[M_t(\beta) \mid \mathcal{F}_{t-1}] &= \mathbb{E}[M_{t-1}(\beta)\, B_t(\beta) \mid \mathcal{F}_{t-1}] \\ &= M_{t-1}(\beta)\, \mathbb{E}\Big[1 + \lambda_t(\tilde{R}_t(\beta) - \theta) \;\Big|\; \mathcal{F}_{t-1}\Big] \\ &= M_{t-1}(\beta)\, \Big(1 + \lambda_t\big(\mathbb{E}[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}] - \theta\big)\Big).\end{aligned} \tag{viii}$$

By Lemma 1, we have:

$$\mathbb{E}[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}] = \mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}]. \tag{ix}$$

Under the null hypothesis of the theorem,

$$\mathbb{E}[D_t(\beta) \mid \mathcal{F}_{t-1}] \leq \theta. \tag{x}$$

Combining Eq. ix and Eq. x gives:

$$\mathbb{E}[\tilde{R}_t(\beta) \mid \mathcal{F}_{t-1}] \leq \theta. \tag{xi}$$

Substituting Eq. xi into Eq. viii, we obtain:

$$\mathbb{E}[M_t(\beta) \mid \mathcal{F}_{t-1}] \leq M_{t-1}(\beta). \tag{xii}$$

Hence, $\{M_t(\beta)\}_{t\geq 1}$ is a non-negative supermartingale. Taking expectations on both sides of Eq. xii and applying the tower property gives:

$$\mathbb{E}[M_t(\beta)] \leq \mathbb{E}[M_{t-1}(\beta)] \leq \cdots \leq \mathbb{E}[M_0(\beta)] = 1. \tag{xiii}$$

Finally, since $\{M_t(\beta)\}_{t\geq 0}$ is a non-negative supermartingale with initial value $M_0(\beta) = 1$, Ville's inequality can be applied directly here. Therefore, for any $\alpha \in (0, 1)$, we have:

$$\Pr\left(\exists t \in \mathbb{N} : M_t(\beta) \geq \alpha^{-1}\right) \leq \alpha. \tag{xiv}$$

Hence Proved. □

F.4.1 Proof of Corollary 1

Proof. For any fixed $\beta \in \mathcal{G}$, Theorem 3 establishes that $\{M_t(\beta)\}_{t\geq 0}$ is a non-negative supermartingale with $\mathbb{E}[M_t(\beta)] \leq 1$. Applying Ville's inequality at the adjusted significance level αw_β gives:

$$\Pr\left(\exists t \in \mathbb{N} : M_t(\beta) \geq \frac{1}{\alpha w_\beta}\right) \leq \alpha w_\beta. \tag{i}$$

To bound the simultaneous probability over the entire grid, we apply the union bound over all $\beta \in \mathcal{G}$:

$$\Pr\left(\bigcup_{\beta\in\mathcal{G}} \left\{\exists t \in \mathbb{N} : M_t(\beta) \geq \frac{1}{\alpha w_\beta}\right\}\right) \leq \sum_{\beta\in\mathcal{G}} \Pr\left(\exists t \in \mathbb{N} : M_t(\beta) \geq \frac{1}{\alpha w_\beta}\right). \tag{ii}$$

Substituting equation i into equation ii and applying the weight constraint gives:

$$\sum_{\beta\in\mathcal{G}} \alpha w_\beta = \alpha \sum_{\beta\in\mathcal{G}} w_\beta \leq \alpha(1) = \alpha. \tag{iii}$$

Hence Proved. □

F.5 Proof of Theorem 4

Proof. We begin by defining the post-change stopping time as follows:

$$\tau_{\alpha,t_0}(\beta) \coloneqq \inf\left\{t \geq t_0 : M_t(\beta) \geq \alpha^{-1}\right\}. \tag{i}$$

Recall that the Edit Risk Martingale satisfies the following condition:

$$M_t(\beta) = M_{t-1}(\beta)\left(1 + \lambda_t(\tilde{R}_t(\beta) - \theta)\right). \tag{ii}$$

The sampling rule and sensitivity condition imply that every post-change factor is at least $1/2$, as established below. Since $M_{t_0-1}(\beta) > 0$, the logarithms are therefore well defined. Taking logarithms gives us:

$$\log M_t(\beta) = \log M_{t-1}(\beta) + \log\left(1 + \lambda_t(\tilde{R}_t(\beta) - \theta)\right). \tag{iii}$$

Hence, for every $t \geq t_0$,

$$\log M_t(\beta) = \log M_{t_0-1}(\beta) + \sum_{i=t_0}^{t} \log\left(1 + \lambda_i(\tilde{R}_i(\beta) - \theta)\right). \tag{iv}$$

For simplicity, define,

$$X_t \coloneqq \lambda_t(\tilde{R}_t(\beta) - \theta). \tag{v}$$

Under the sampling rule in Eq. 11, $q_t < 1$ implies $\hat{r}_t \leq \theta q_t$. Therefore, if $K_t = 1$ and $q_t < 1$, we have:

$$\tilde{R}_t(\beta) = \frac{D_t(\beta)}{q_t} - \frac{1 - q_t}{q_t}\hat{r}_t \geq -\theta(1 - q_t) \geq -\theta(1 - q_{\min}).$$

If $K_t = 0$, then $\tilde{R}_t(\beta) = \hat{r}_t \geq 0$. If $q_t = 1$, then $K_t = 1$ almost surely and $\tilde{R}_t(\beta) = D_t(\beta) \geq 0$. Thus, $\tilde{R}_t(\beta) \geq -\theta(1 - q_{\min})$ in all cases. Using $\lambda_t \leq \bar{\lambda}$ and the theorem condition $\bar{\lambda}\theta(2 - q_{\min}) \leq 1/2$, we obtain:

$$X_t \geq -\bar{\lambda}\theta(2 - q_{\min}) \geq -\frac{1}{2}.$$

Applying the bound $\log(1 + x) \geq x - x^2$ for $x \geq -1/2$ gives

$$\log\left(1 + \lambda_t(\tilde{R}_t(\beta) - \theta)\right) \geq \lambda_t(\tilde{R}_t(\beta) - \theta) - \lambda_t^2(\tilde{R}_t(\beta) - \theta)^2. \tag{vi}$$

Taking conditional expectation with respect to $\mathcal{F}_{t-1}$, and using the fact that $\mathbb{E}[\tilde{R}_t(\beta) - \theta \mid \mathcal{F}_{t-1}] \geq \epsilon$, we get:

$$\begin{aligned}\mathbb{E}\Big[\log\left(1 + \lambda_t(\tilde{R}_t(\beta) - \theta)\right)\Big|\mathcal{F}_{t-1}\Big] &\geq \lambda_t\,\mathbb{E}[\tilde{R}_t(\beta) - \theta \mid \mathcal{F}_{t-1}] - \lambda_t^2\,\mathbb{E}\Big[(\tilde{R}_t(\beta) - \theta)^2 \mid \mathcal{F}_{t-1}\Big] \\ &\geq \lambda_{\min}\epsilon - \bar{\lambda}^2\bar{v} = \delta_{\text{drift}}.\end{aligned} \tag{vii}$$

We now define the stopped log-process for $t \geq t_0 - 1$ as:

$$Y_t \coloneqq \log\frac{M_{\tau_{\alpha,t_0}(\beta)\wedge t}(\beta)}{M_{t_0-1}(\beta)} - \delta_{\text{drift}}\left((\tau_{\alpha,t_0}(\beta) \wedge t) - t_0 + 1\right). \tag{viii}$$

By Eq. iv and the drift lower bound, $\{Y_t\}_{t \geq t_0-1}$ is a submartingale with $Y_{t_0-1} = 0$, yielding $\mathbb{E}[Y_t \mid \mathcal{F}_{t_0-1}] \geq 0$. Expanding this gives us:

$$\mathbb{E}\left[\log\frac{M_{\tau_{\alpha,t_0}(\beta)\wedge t}(\beta)}{M_{t_0-1}(\beta)}\;\middle|\;\mathcal{F}_{t_0-1}\right] - \delta_{\text{drift}}\,\mathbb{E}[(\tau_{\alpha,t_0}(\beta) \wedge t) - t_0 + 1 \mid \mathcal{F}_{t_0-1}] \geq 0. \tag{ix}$$

Since $\tilde{R}_t(\beta) \leq q_{\min}^{-1}$ in both sampling branches, the multiplicative factor is at most $1 + \bar{\lambda}(q_{\min}^{-1} - \theta)$. Before the stopping time, the monitor is below α^{-1}, and at the stopping time it can exceed this threshold by at most this factor. Consequently, for every finite $t \geq t_0$,

$$\log\frac{M_{\tau_{\alpha,t_0}(\beta)\wedge t}(\beta)}{M_{t_0-1}(\beta)} \leq \log\alpha^{-1} - \log M_{t_0-1}(\beta) + \log\left(1 + \bar{\lambda}\left(\frac{1}{q_{\min}} - \theta\right)\right).$$

Substituting this bound into Eq. ix and applying conditional monotone convergence to $(\tau_{\alpha,t_0}(\beta) \wedge t) - t_0 + 1$ as $t \to \infty$ finally gives us:

$$\mathbb{E}[\tau_{\alpha,t_0}(\beta) - t_0 + 1 \mid \mathcal{F}_{t_0-1}] \leq \frac{\log\alpha^{-1} - \log M_{t_0-1}(\beta) + \log\left(1 + \bar{\lambda}\left(\frac{1}{q_{\min}} - \theta\right)\right)}{\delta_{\text{drift}}}. \tag{x}$$

The right-hand side is finite almost surely, which also implies $\tau_{\alpha,t_0}(\beta) < \infty$ almost surely. This proves the theorem. □

F.5.1 PROOF OF COROLLARY 2

Proof. Let $Z \coloneqq \tau_{\alpha,t_0}(\beta) - t_0 + 1$. Since $\tau_{\alpha,t_0}(\beta) \geq t_0$, Z is non-negative. Applying Markov's inequality conditionally on $\mathcal{F}_{t_0-1}$ gives

$$\Pr\left(Z > h \mid \mathcal{F}_{t_0-1}\right) \leq \frac{\mathbb{E}[Z \mid \mathcal{F}_{t_0-1}]}{h}. \tag{i}$$

Substituting the upper bound from Theorem 4 into Eq. i results in:

$$\Pr\left(\tau_{\alpha,t_0}(\beta) > t_0 + h - 1 \mid \mathcal{F}_{t_0-1}\right) \leq \frac{\log \alpha^{-1} - \log M_{t_0-1}(\beta) + \log\left(1 + \bar{\lambda}\left(\frac{1}{q_{\min}} - \theta\right)\right)}{\delta_{\text{drift}}\, h}. \tag{ii}$$

This proves the corollary. □