Online Safety Monitoring for LLMs

Mona Schirmer [1] **Metod Jazbec** [1] **Alexander Timans** [1] **Christian Naesseth** [1] **Maja Waldron** [2] **Eric Nalisnick** [3]

Abstract

Despite alignment training, LLMs remain prone to generating unsafe outputs at deployment time. Monitoring outputs online and raising an alarm when safety can no longer be assumed is therefore critical. We study a simple real-time monitor that turns a verifier signal from an external model into an alarm decision by thresholding, with the threshold calibrated via risk control. In experiments on mathematical reasoning and red teaming datasets, we show that this simple design is competitive with more advanced monitors based on sequential hypothesis testing.

1. Introduction

Large Language Models (LLMs) have become integrated into our everyday lives as search engines (Jin et al., 2025; Xiong et al., 2024), coding assistants (Zhao et al., 2023), and companions (Zhang et al., 2025a). As their applicability grows, so does the potential harm caused by malicious LLM outputs. Despite remarkable performance across a wide range of tasks, LLMs remain prone to generating hallucinated, factually incorrect (Ravichander et al., 2025), or harmful output (Yu et al., 2025) when deployed.

LLM safety research has addressed these challenges using better alignment strategies during training (Bai et al., 2022), and rigorous offline evaluation before deploying (Wang et al., 2023). Nevertheless, these pre-deployment safety measures cannot account for all possible prompt scenarios, and the risk of harmful outputs remains. There has thus been substantial effort into developing guardrails at inference time (Inan et al., 2023; Sharma et al., 2025; Baker et al., 2025) – though many of them are designed for post-hoc detection. We argue that such monitors should be able to operate in an online stream setting, detecting unsafe content as it is produced, and stopping harmful generations in real-time (Wang et al., 2025; Li et al., 2025).

[1]UvA Bosch-Delta Lab, University of Amsterdam [2]University of Wisconsin Madison [3]Johns Hopkins University. Correspondence to: Mona Schirmer <m.c.schirmer@uva.nl>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

The inherent challenges of the online monitoring setting are the unavailability of safety labels, and the fast reaction time required for effective detection. In an ideal setting, every output is reviewed by a group of human experts who judge whether it is harmful, incorrect, or otherwise problematic. In the real world, however, we only get an *approximate* signal that informs us of the safety of the current output, such as the predictive probability from a safeguard classifier. The task of an online monitor is then to translate this signal into a binary decision—raise an alarm or remain silent—as the LLM produces its output. The monitor itself comes with two main risks: *(i)* raising false alarms unnecessarily interrupts user experience and functionality of the LLM, and *(ii)* failing to detect true alarms gives a sense of false security and renders the monitor useless.

In this paper, we study the online monitoring setting covering several safety risks (Weidinger et al., 2023): factual correctness, toxicity and malicious use. We deploy a simple statistical framework based on *risk control* (Angelopoulos et al., 2022) that converts any safety signal into a binary decision rule, and offers statistical guarantees on the false alarm or missed detection rate. The framework is universally applicable to different monitoring purposes and can leverage arbitrary proxy signals. Through experiments on mathematical problem solving and red teaming conversations, we show that our simple approach is competitive with more involved methods (Sadhuka et al., 2025), while detecting failures earlier in the generation process.

2. Problem Setting

We consider the problem of monitoring the safety of an LLM's output as it unfolds. Let $t = 0, 1, \dots$ be a time index. At $t = 0$, the LLM is given a user prompt $x \sim P_x$ sampled from a prompt distribution. The LLM then produces an output sequence $o_{1:T} = (o_1, \dots, o_T)$ of variable length T, where each o_t is sampled autoregressively from the model's generative distribution $P_\theta(o_t \mid x, o_{1:t-1})$. Depending on the use case, each o_t may represent a token of the output text, a step in the reasoning chain, or a response of the LLM to the user. Let $y \in \mathcal{Y} = \{0, 1\}$ be the safety variable, sampled from $P(y \mid o_{1:T})$, which captures the safety of the output sequence, with $y = 1$ indicating a safe state (e.g. correct, harmless output) and $y = 0$ indicating an unsafe state (e.g.

incorrect, harmful output). The goal of an online monitor is to identify an unsafe sequence as early as possible, while continuously inspecting $o_{1:t}$ as it unfolds.

3. Online Monitoring via Risk Control

We study a simple statistical baseline for monitoring LLM safety at inference time. More formally, given a stream of safety signals $s_{1:t}$, we seek a decision rule – i.e., a well-calibrated threshold λ – that maps $s_{1:t}$ to a binary judgment of whether the output is safe. We discuss appropriate choices of safety signals in § 3.1 and state the monitor function in § 3.2. In § 3.3, we define the risks of the monitor we wish to control and lastly, in § 3.4, provide two options on how to determine a decision threshold λ which guarantees these risks remain controlled *in expectation* or *with high probability*.

3.1. Safety Signal

To assess the safety of an output, we rely on an imperfect signal s_t that carries information about y. For example, s_t may be the predictive probability of an external verifier model $p_\psi(y \mid x, o_{1:t})$ that assesses safety given all currently observed outputs $o_{1:t}$. In mathematical reasoning, p_ψ may correspond to a process reward model (PRM) (Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2025b); in content moderation, it can be an LLM safeguard (Inan et al., 2023; Zeng et al., 2024). While external models explicitly trained for this binary prediction are effective, they are also potentially expensive to deploy alongside the LLM generator. Internal signals from the generating LLM $p_\theta(o_t \mid x, o_{1:t-1})$ have therefore been explored as a cheaper alternative (Agarwal et al., 2026; Kossen et al., 2024). We conduct an ablation on this trade-off in § 4.3.

3.2. Safety Monitor

The monitor processes the signal sequence $s_{1:t}$ online and emits a binary decision at each time step $t = 1, 2 \ldots$, with $\Phi_t = 1$ indicating the output has been flagged as unsafe. We define it as a stopping rule governed by a single threshold λ: the monitor raises an alarm at the first step k at which s_k falls below λ. Formally, we write

$$\Phi_t := \mathbf{1}\{\exists k,\ 1 \leq k \leq t : s_k < \lambda\}. \tag{1}$$

Upon raising an alarm at time t, downstream interventions may be triggered, for instance, by halting generation, escalating to a stronger verifier, or invoking a human-in-the-loop review. We emphasize the simplicity of the rule in Eq. (1): it relies on a single, time-invariant threshold λ applied uniformly across all signals s_t. How to choose λ such that the monitor's risks are provably controlled is the subject of the remainder of this section. First, we turn to formalizing our notions of risk themselves.

3.3. Risks of Monitors

As discussed in § 1, a monitor is exposed to two complementary error modes, mirroring the type I and type II errors of classical hypothesis testing (Kaur & Stoltzfus, 2017). The *false alarm risk* captures the probability of flagging a sequence that is in fact safe (type I error, $\mathcal{R}^I$), while the *missed detection risk* captures the probability of failing to raise an alarm on a sequence that is unsafe (type II error, $\mathcal{R}^{II}$). We focus on the false alarm risk henceforth and refer to § B for methodology and experiments on the missed detection risk.

False Alarm Risk. The probability that the monitor governed by a specific λ raises an alarm at some step t given that the underlying sequence is safe is denoted

$$\mathcal{R}^I(\lambda) = \mathbb{P}(\exists\, t \geq 1 : s_t < \lambda \mid y = 1). \tag{2}$$

Note that this is an expectation of the binary loss $\ell(s_{1:t}, \lambda) = \mathbf{1}\{\exists\, t \geq 1 : s_t < \lambda\}$ taken over safe samples only ($y = 1$), so $\mathcal{R}^I(\lambda) \in [0, 1]$. $\mathcal{R}^I(\lambda)$ is *monotonically increasing* in λ: raising the threshold makes the monitor more eager to flag, increasing false alarms.

3.4. Controlling Monitoring Risks

Having defined the risks of interest in § 3.3, we now turn to the central calibration question: how should the threshold λ be chosen so that the resulting monitor provably controls a targeted risk $\mathcal{R} \in \{\mathcal{R}^I, \mathcal{R}^{II}\}$ at user-specified risk level $\epsilon \in (0, 1)$? We consider access to a labeled held-out calibration dataset $\mathcal{D}_{\text{cal}} = \{(x^{(i)}, o^{(i)}_{1:T}, y^{(i)})\}_{i=1}^n$ of n calibration samples, drawn exchangeably from the same distribution as the deployment data. We can then evaluate empirical risks $\hat{\mathcal{R}}(\lambda; \mathcal{D}_{\text{cal}})$ for any candidate threshold $\lambda \in \Lambda$. Two notions of control are natural in this setting, yielding two distinct calibration procedures that differ in the strength of their statistical guarantees.

Control in expectation. The first option, *conformal risk control* (Angelopoulos et al., 2022), provides a way to find a threshold $\hat{\lambda}$ on a given calibration dataset D_{cal}, such that the risk, i.e. expected loss, of future test points is bounded *on average* over draws of possible calibration sets:

$$\mathbb{E}_{\mathcal{D}_{\text{cal}}}[\mathcal{R}(\hat{\lambda})] \leq \epsilon. \tag{3}$$

Exploiting that the loss ℓ is non-decreasing in λ for Eq. (2), the guarantee is achieved by selecting the largest threshold whose finite-sample-corrected empirical risk still lies below the target level:

$$\hat{\lambda}_{\text{CRC}} := \max\left\{\lambda \in \Lambda \,:\, \frac{n}{n+1}\,\hat{\mathcal{R}}(\lambda; \mathcal{D}_{\text{cal}}) + \frac{1}{n+1} \leq \epsilon\right\}. \tag{4}$$

Control with high probability. The second option, based on Bates et al. (2021), provides a stronger guarantee: for a user-specified confidence level $(1-\delta)$ the same risk is bounded for *all but a δ-fraction of calibration draws*, formally

$$\mathbb{P}_{\mathcal{D}_{\text{cal}}}\Big(\mathcal{R}(\hat{\lambda}) \leq \epsilon\Big) \geq 1-\delta. \tag{5}$$

The construction proceeds in two steps. First, a $(1-\delta)$ upper confidence bound (UCB) $U(\lambda, n, \delta)$ on the empirical risk $\hat{\mathcal{R}}(\lambda; \mathcal{D}_{\text{cal}})$ is computed on the calibration set. It follows $\mathbb{P}_{\mathcal{D}_{\text{cal}}}(\mathcal{R}(\lambda) \leq U(\lambda, n, \delta)) \geq 1-\delta$ for all $\lambda \in \Lambda$. Then, the decision threshold is taken as the largest value for which this upper bound still meets the target rate ϵ, hence

$$\hat{\lambda}_{\text{UCB}} := \max\{\lambda \in \Lambda \,:\, U(\lambda, n, \delta) \leq \epsilon\}. \tag{6}$$

Because the true risk $\mathcal{R}(\lambda)$ lies below the bound with probability at least $(1-\delta)$, any λ selected by this rule inherits the same guarantee, yielding high-probability control. We use the Hoeffding-Bentkus bound (Bentkus, 2004) as UCB.

In contrast to in-expectation control, high-probability control offers a stronger deployment guarantee at the cost of a more conservative threshold and, typically, the need for a larger calibration set size n. Thus, in practice, UCB may be preferred when the cost of a single safety violation is high and sufficient calibration samples are available. We refer to the monitor instantiated by Eq. (4) as CRC and Eq. (6) as UCB respectively.

4. Experiments

We evaluate the risk-controlling monitor on two safety use cases: (i) factuality by monitoring step-level correctness for mathematical reasoning (§ 4.1) and (ii) malicious use and toxicity, by monitoring multi-turn conversations with users (§ 4.2). Finally, § 4.3 studies the cost–performance trade-off when using cheap internal model signals in place of external verifiers. Our code is available at `https://github.com/monasch/llm-monitor`.

Baselines We compare the simple risk control monitor to two versions of the e-valuator (Sadhuka et al., 2025). All monitors observe the same signal s_t and have access to the same calibration set, but differ in the monitor function. Importantly, instead of thresholding the raw signal by a single λ (Eq. (1)), e-valuators learn T density estimators (one for each step t) on the signal sequence $s_{1:T}$. The resulting *e-process* is used as evidence signal for a sequential hypothesis test. Details on e-valuators are in § C.

Metrics We assess the monitors across 3 metrics. Over $i = 1, \ldots, N$ test sequences, let $\mathcal{I}_{\text{s}} = \{i : y^{(i)} = 1\}$, $\mathcal{I}_{\text{us}} = \{i : y^{(i)} = 0\}$, and $\mathcal{I}_{\Phi} = \{i : \exists t, \Phi_t^{(i)} = 1\}$ denote the safe, unsafe, and flagged sequences, respectively.

- **False Alarm Rate**, the share of safe sequences that are incorrectly flagged, $\hat{R}^{I}(\hat{\lambda}) = |\mathcal{I}_{\Phi} \cap \mathcal{I}_{\text{s}}| \,/\, |\mathcal{I}_{\text{s}}|$, i.e., the empirical estimate of the false alarm risk in Eq. (2). Lower is better.
- **Power**, the share of unsafe sequences that are correctly flagged, $1 - \hat{R}^{II}(\hat{\lambda}) = |\mathcal{I}_{\Phi} \cap \mathcal{I}_{\text{us}}| \,/\, |\mathcal{I}_{\text{us}}|$, i.e., one minus the empirical estimate of the missed detection risk (Eq. (7),§ B). Higher is better.
- **Detection Delay**, the average fraction of steps the model processes before an alarm is correctly raised, $\hat{DD} = \frac{1}{|\mathcal{I}_{\Phi} \cap \mathcal{I}_{\text{us}}|} \sum_{i \in \mathcal{I}_{\Phi} \cap \mathcal{I}_{\text{us}}} t_{\min}^{(i)} / T^{(i)}$, where $t_{\min}^{(i)} = \min\{t : s_t^{(i)} < \hat{\lambda}\}$ is the alarm time and $T^{(i)}$ is the sequence length. Lower is better.

4.1. Monitoring for Factuality

In our first experiment, we consider the safety risk caused by LLMs providing factually incorrect statements (Weidinger et al., 2021; Ji et al., 2023). We focus on mathematical reasoning, where incorrect solutions can cause false belief in the user especially when they overestimate LLM competence (Weidinger et al., 2021; Steyvers et al., 2025).

Setup We use the MATH dataset (Hendrycks et al., 2021) to monitor an LLM's capacity to produce correct step-by-step solutions to mathematical problems. We employ two generating LLMs of varying problem-solving capacity: Claude Haiku 4.5 (Anthropic, 2025) solves 90% of problems correctly, whereas Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) solves only 26%. We use OpenAI's o3-mini (OpenAI, 2025) to compare each generator's final response with the ground-truth solution, providing the y labels. As signal s_t, we use the step-wise output probability of Qwen2.5-Math-PRM-7B (Zhang et al., 2025b).

Risk remains empirically controlled. Fig. 1 shows the false alarm rate, power, and detection delay when controlling the monitor's false alarm risk (Eq. (2)) across varying tolerance levels ϵ. The monitors with high-probability risk control (Eq. (5)) – UCB, e-valuator-anytime, and e-valuator-PAC – successfully control the false alarm rate (*first row*), with the $1-\delta$ confidence interval over 10 runs (*shaded region*) lying below the identity line; the only exception is e-valuator-anytime (—), which violates the bound at $\epsilon \in \{0.05, 0.1\}$ on the Mistral model. The CRC monitor likewise satisfies its in-expectation guarantee (—< - -).

Risk controlling monitors raise alarms earlier. Despite their simplicity, CRC and UCB are surprisingly on par with the more complex e-valuator monitors. Looking at power (*second row*, *how many* incorrect sequences are detected), e-valuator-anytime has the highest at small tolerance levels, though this comes at the cost of risk violation. E-valuator-PAC and UCB are on par. As expected, the less conserva-

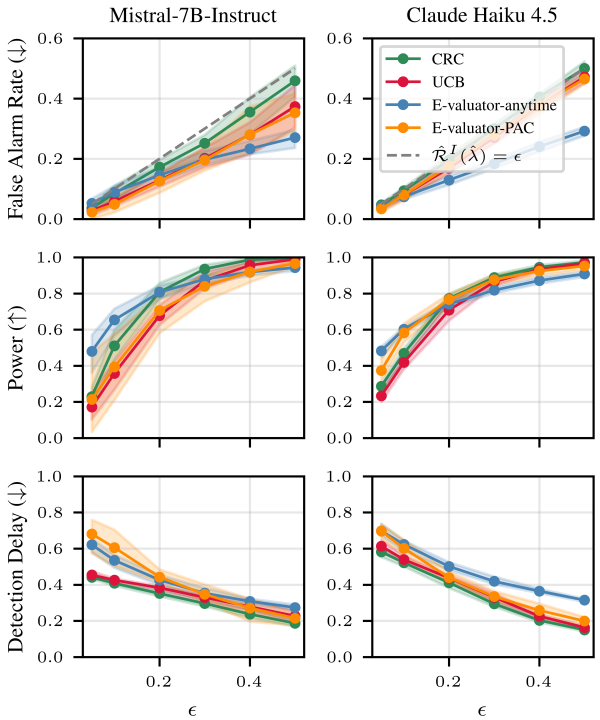

Figure 1. Monitoring factuality on mathematical reasoning (MATH): CRC and UCB calibrate a single threshold to turn a PRM score into an alarm, yet detect incorrect answers earlier (*third row*) than e-valuators, which train several density estimators.

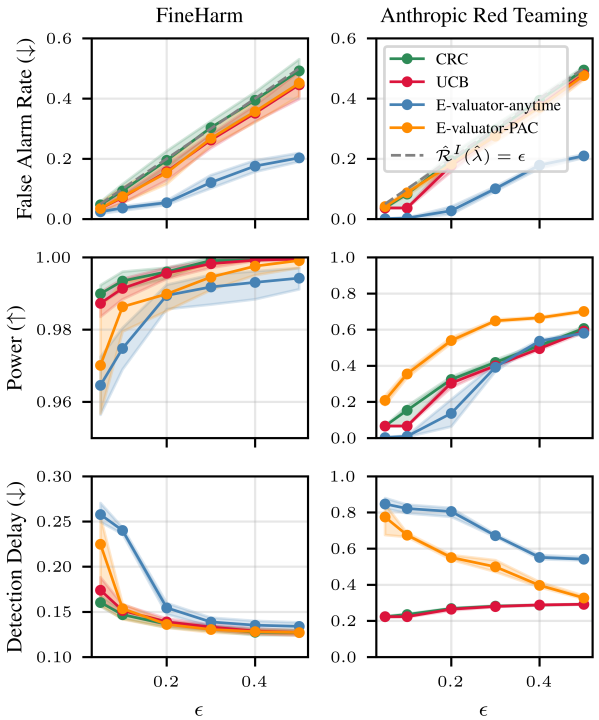

Figure 2. Monitoring harmlessness: All monitors maintain false alarm control (*first row*); power (*second row*) varies strongly based on whether the safeguard verifier is trained for the specific online detection (FineHarm, *first column*) or not (Red Teaming, *second column*). Despite its simplicity, risk controlling monitors (CRC, UCB) detect harmful output earlier (*third row*).

tive CRC yields higher power than UCB. Turning to detection delay (*third row*, *how quickly* the detected sequences are flagged) yields an interesting picture: although the e-valuator monitors detect more incorrect sequences, they do so much later. CRC and UCB instead flag incorrect sequences after about half the sequence, with CRC slightly ahead of UCB. Earlier detection is desirable in practice, as it limits the user's exposure to harmful outputs and reduces the token-generation cost.

4.2. Monitoring for Harmlessness

In our next experiment, unsafety stems not from incorrect outputs, but from the LLM complying with malicious requests or producing discriminatory and toxic content.

Setup We evaluate our monitor on two datasets of harmful LLM interactions. The Anthropic Red Teaming data (Ganguli et al., 2022) consists of multi-turn conversations between red-team members and an LLM, each rated for attack success, which provides the sequence-level safety label y. At each turn t, we feed the conversation up to the current time point, $o_{1:t}$, to Llama Guard (Inan et al., 2023) and use its predicted probability of the `safe` token as the verifier signal s_t. The FineHarm data (Li et al., 2025) is built on WildGuard (Han et al., 2024) and WildJailbreak (Jiang et al., 2024) and comprises single-turn interactions. Here we use their SCM verifier, Qwen2.5-1.5B fine-tuned for token-level harmfulness detection, and take its predicted probability of safety given the tokens up to t as signal s_t.

Better signals, better monitors. Fig. 2 displays results for FineHarm (first column) and Anthropic Red Teaming (second column). We highlight four findings. First, $\hat{R}^I$ remains controlled across monitors and datasets. Second, e-valuator-PAC achieves the highest power on Red Teaming, while CRC and UCB do so on FineHarm. Third, detection delay is again lower for the risk-controlling threshold methods. Fourth, comparing the two datasets highlights an important point: the monitor can only be as good as its verifier signal. On FineHarm, the SCM verifier is fine-tuned on the training set for token-level harmfulness, yielding strong separability (power close to 1), whereas Llama Guard's power on Red Teaming stagnates below 0.8. This underscores the need for accurate signals tailored to the use case.

4.3. Trading Monitor Efficiency for Performance

Running an external verifier alongside the generator may be too expensive in a practical deployment setting: it adds

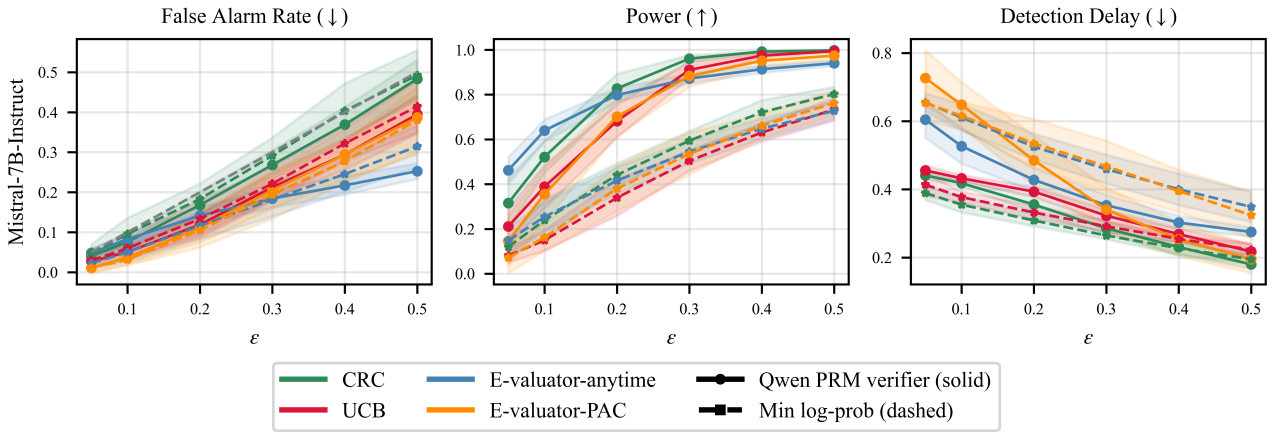

Figure 3. Signal ablation of token log-probabilities vs. external PRM: False alarm rate, power, and detection delay on Mistral-7B-Instruct (MATH) across target levels ε; solid curves use the Qwen PRM, dashed curves use the per-step minimum token log-probability. The free log-prob signal yields substantially lower power across all four monitors.

a second forward pass at every step. A natural question is therefore whether a much cheaper signal — one the generator already produces for free — can substitute for it, and we examine this in our next experiment.

Set-up. We run a signal ablation on Mistral-7B-Instruct reasoning chains on the MATH dataset. In place of the Qwen PRM score, we use the generator's own token log-probabilities[1] as the safety signal. Unlike the external verifier, this signal incurs practically no additional computational cost at inference time. Concretely, we found the per-step minimum over token log-probabilities to be the most informative aggregation. Formally, let w_j denote the j-th output token of the generated chain, so that $\log p_\theta(w_j \mid w_{1:j-1}, x)$ is the generator's log-probability of that token given all preceding tokens and the prompt x. For a step k spanning tokens $[j^{(k)}_{\text{start}}, j^{(k)}_{\text{end}}]$, we set $s_k = \min_{j \in [j^{(k)}_{\text{start}}, j^{(k)}_{\text{end}}]} \log p_\theta(w_j \mid w_{1:j-1}, x)$, so that steps containing an unusually uncertain token receive a more negative (more unsafe) score. We feed this signal into the same set of monitors and sweep the target false-alarm level $\varepsilon \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

Token log-probabilities are a weaker signal than an external PRM. Fig. 3 reports false alarm rate, power, and detection delay for the log-prob signal (dashed curves) against the PRM-based runs (solid curves). The middle column makes the trade-off between verifier cost and signal strength explicit: at matched false-alarm rates, the PRM-based monitors achieve substantially higher power than their log-prob counterparts — for example, near $\varepsilon = 0.3$ the PRM variants already exceed 0.9 power while the log-prob counterparts sit around 0.5. The generator's own token-level confidence is therefore a meaningfully weaker correctness signal than a dedicated PRM, and the gap is large enough that the cost savings come at a real monitoring-quality cost.

[1]Besides token log-probabilities, we also considered measures that capture the full token distribution at every step, e.g., self-certainty (Kang et al., 2026), but found them to perform on-par with the (simpler) token log-probability signal in our experiments.

5. Conclusion, Limitations and Future Work

We argue for monitoring LLMs in real-time, enabling intervention as soon as safety can no longer be ensured. We compared statistical frameworks providing guarantees of differing strength, and found that calibrating a single threshold on a proxy signal is a simple yet effective approach to obtaining monitoring guarantees during deployment.

Limitations and Future Work. Calibrating a single time-invariant threshold on a verifier signal is attractive for deployment: it adds negligible computational overhead (e.g. no additional density estimator required) and imposes light restrictions on the calibration data (e.g. no sufficient coverage of varying length sequence necessary). However, it has two key limitations. First, the monitor is only as good as its signal — inheriting the verifier's limitations in terms of informativeness, deployment cost, and adversarial robustness. Future work could address these limitations by (i) combining multiple signals into a more informative and robust statistic; (ii) identifying the best accuracy-cost trade-off for a given safety risk (Gui et al., 2024; Kaddour et al., 2026); or (iii) issuing additional targeted safety checks once an alarm is triggered. Second, it ignores temporal structure in the signal, since the verifier score at step t may systematically depend on t. Future work may (iv) calibrate a per-step threshold using more advanced procedures such as Pareto testing (Laufer-Goldshtein et al., 2022).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, S., Zhang, Z., Yuan, L., Han, J., and Peng, H. The unreasonable effectiveness of entropy minimization in llm reasoning. *Advances in Neural Information Processing Systems*, 38:107150–107180, 2026.

Amoukou, S. I., Bewley, T., Mishra, S., Lecue, F., Magazzeni, D., and Veloso, M. Sequential harmful shift detection without labels. *Advances in Neural Information Processing Systems*, 37:129279–129302, 2024.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

Anthropic. System card: Claude haiku 4.5. Technical report, Anthropic, October 2025. URL https://assets.anthropic.com/m/99128ddd009bdcb/Claude-Haiku-4-5-System-Card.pdf.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Baker, B., Huizinga, J., Gao, L., Dou, Z., Guan, M. Y., Madry, A., Zaremba, W., Pachocki, J., and Farhi, D. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.

Bar, Y., Shaer, S., and Romano, Y. Protected test-time adaptation via online entropy matching: A betting approach. *Advances in Neural Information Processing Systems*, 2024.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

Bentkus, V. On hoeffding's inequalities. 2004.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Cherian, J. J., Gibbs, I., and Candès, E. J. Large language model validity via enhanced conformal prediction methods. *Advances in Neural Information Processing Systems*, 37:114812–114842, 2024.

Chuang, Y.-S., Qiu, L., Hsieh, C.-Y., Krishna, R., Kim, Y., and Glass, J. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, 2024.

Davidov, H., Feldman, S., Freidkin, G., and Romano, Y. Calibrated predictive lower bounds on time-to-unsafe-sampling in llms. *arXiv preprint arXiv:2506.13593*, 2025.

Feldman, S. and Romano, Y. How many iterations to jailbreak? dynamic budget allocation for multi-turn llm evaluation. *arXiv preprint arXiv:2605.06605*, 2026.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Greenblatt, R., Shlegeris, B., Sachan, K., and Roger, F. Ai control: Improving safety despite intentional subversion. *arXiv preprint arXiv:2312.06942*, 2023.

Gui, Y., Jin, Y., and Ren, Z. Conformal alignment: Knowing when to trust foundation models with guarantees. *Advances in Neural Information Processing Systems*, 37: 73884–73919, 2024.

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., and Dziri, N. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in neural information processing systems*, 37:8093–8131, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Jazbec, M., Timans, A., Veljković, T. H., Sakmann, K., Zhang, D., Naesseth, C. A., and Nalisnick, E. Fast yet safe: Early-exiting with risk control. *Advances in Neural Information Processing Systems*, 37:129825–129854, 2024.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. arxiv. *arXiv preprint arXiv:2310.06825*, 10:3, 2023.

Jiang, L., Rao, K., Han, S., Ettinger, A., Brahman, F., Kumar, S., Mireshghallah, N., Lu, X., Sap, M., Choi, Y., and Dziri, N. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models, 2024. URL https://arxiv.org/abs/2406.18510.

Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., and Han, J. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Kaddour, J., Patel, S., Dovonon, G., Richter, L., Minervini, P., and Kusner, M. J. Agentic uncertainty reveals agentic overconfidence. *arXiv preprint arXiv:2602.06948*, 2026.

Kang, Z., Zhao, X., and Song, D. Scalable best-of-n selection for large language models via self-certainty. *Advances in neural information processing systems*, 38: 19720–19745, 2026.

Kaur, P. and Stoltzfus, J. Type i, ii, and iii statistical errors: A brief overview. *International Journal of Academic Medicine*, 2017.

Korbak, T., Balesni, M., Barnes, E., Bengio, Y., Benton, J., Bloom, J., Chen, M., Cooney, A., Dafoe, A., Dragan, A., et al. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*, 2025.

Kossen, J., Han, J., Razzak, M., Schut, L., Malik, S., and Gal, Y. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Laufer-Goldshtein, B., Fisch, A., Barzilay, R., and Jaakkola, T. Efficiently controlling multiple risks with pareto testing. *arXiv preprint arXiv:2210.07913*, 2022.

Li, Y., Sheng, Q., Yang, Y., Zhang, X., and Cao, J. From judgment to interference: Early stopping llm harmful outputs via streaming content monitoring. *arXiv preprint arXiv:2506.09996*, 2025.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Manakul, P., Liusie, A., and Gales, M. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Markov, T., Zhang, C., Agarwal, S., Nekoul, F. E., Lee, T., Adler, S., Jiang, A., and Weng, L. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 15009–15018, 2023.

Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Mohri, C. and Hashimoto, T. Language models with conformal factuality guarantees. *arXiv preprint arXiv:2402.10978*, 2024.

OpenAI. Openai o3-mini system card. Technical report, OpenAI, January 2025. URL https://cdn.openai.com/o3-mini-system-card-feb10.pdf.

Podkopaev, A. and Ramdas, A. Tracking the risk of a deployed model and detecting harmful distribution shifts. *arXiv preprint arXiv:2110.06177*, 2021.

Prinster, D., Han, X., Liu, A., and Saria, S. WATCH: Adaptive monitoring for AI deployments via weighted-conformal martingales. In *International Conference on Machine Learning*, 2025.

Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. Conformal language modeling. *arXiv preprint arXiv:2306.10193*, 2023.

Ramdas, A. and Wang, R. Hypothesis testing with e-values. *Foundations and Trends® in Statistics*, 1(1-2):1–390, 2025.

Ramdas, A., Grünwald, P., Vovk, V., and Shafer, G. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Ravichander, A., Ghela, S., Wadden, D., and Choi, Y. Halogen: Fantastic llm hallucinations and where to find them. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1402–1425, 2025.

Sadhuka, S., Prinster, D., Fannjiang, C., Scalia, G., Regev, A., and Wang, H. E-valuator: Reliable agent verifiers with sequential hypothesis testing. *arXiv preprint arXiv:2512.03109*, 2025.

Schirmer, M., Jazbec, M., Naesseth, C. A., and Nalisnick, E. Monitoring risks in test-time adaptation. *arXiv preprint arXiv:2507.08721*, 2025.

Sharma, M., Tong, M., Mu, J., Wei, J., Kruthoff, J., Goodfriend, S., Ong, E., Peng, A., Agarwal, R., Anil, C., et al. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming. *arXiv preprint arXiv:2501.18837*, 2025.

Shin, J., Ramdas, A., and Rinaldo, A. E-detectors: A nonparametric framework for sequential change detection. *The New England Journal of Statistics in Data Science*, 2023.

Steyvers, M., Tejeda, H., Kumar, A., Belem, C., Karny, S., Hu, X., Mayer, L. W., and Smyth, P. What large language models know and what people think they know. *Nature Machine Intelligence*, 7(2):221–231, 2025.

Timans, A., Verma, R., Nalisnick, E., and Naesseth, C. A. On continuous monitoring of risk violations under unknown shift. *arXiv preprint arXiv:2506.16416*, 2025.

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., et al. Decodingtrust: A comprehensive assessment of trustworthiness in {GPT} models. 2023.

Wang, H., Poskitt, C. M., Sun, J., and Wei, J. Pro2guard: Proactive runtime enforcement of llm agent safety via probabilistic model checking. *arXiv preprint arXiv:2508.00500*, 2025.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Wang, X., Suresh, A., Zhang, A., More, R., Jurayj, W., Van Durme, B., Farajtabar, M., Khashabi, D., and Nalisnick, E. Conformal thinking: Risk control for reasoning on a compute budget. *arXiv preprint arXiv:2602.03814*, 2026.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Weidinger, L., Rauh, M., Marchal, N., Manzini, A., Hendricks, L. A., Mateos-Garcia, J., Bergman, S., Kay, J., Griffin, C., Bariach, B., et al. Sociotechnical safety evaluation of generative ai systems. *arXiv preprint arXiv:2310.11986*, 2023.

Wu, M., Zhou, C., Bates, S., and Jaakkola, T. Thought calibration: Efficient and confident test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14302–14316, 2025.

Xiong, H., Bian, J., Li, Y., Li, X., Du, M., Wang, S., Yin, D., and Helal, S. When search engine services meet large language models: visions and challenges. *IEEE Transactions on Services Computing*, 17(6):4558–4577, 2024.

You, W., Xue, A., Havaldar, S., Rao, D., Jin, H., Callison-Burch, C., and Wong, E. Probabilistic soundness guarantees in llm reasoning chains. *Conference on Empirical Methods in Natural Language Processing*, 2025.

Yu, M., Meng, F., Zhou, X., Wang, S., Mao, J., Pan, L., Chen, T., Wang, K., Li, X., Zhang, Y., et al. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6216–6226, 2025.

Zeng, H., Huang, J., Jing, B., Wei, H., and An, B. Pac reasoning: Controlling the performance loss for efficient reasoning. *arXiv preprint arXiv:2510.09133*, 2025.

Zeng, W., Liu, Y., Mullins, R., Peran, L., Fernandez, J., Harkous, H., Narasimhan, K., Proud, D., Kumar, P., Radharapu, B., et al. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*, 2024.

Zhang, Y., Zhao, D., Hancock, J. T., Kraut, R., and Yang, D. The rise of ai companions: how human-chatbot relationships influence well-being. *arXiv preprint arXiv:2506.12605*, 2025a.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025b.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124, 2023.

A. Related Work

LLM Monitoring Various efforts (Weidinger et al., 2021, 2023; Bommasani et al., 2021) have stressed the importance of monitoring LLMs at deployment time, for factuality (Manakul et al., 2023; Chuang et al., 2024), content moderation (Inan et al., 2023; Zeng et al., 2024) or against obfuscation (Korbak et al., 2025; Baker et al., 2025; Greenblatt et al., 2023). For content moderation, for instance, monitoring tools are typically taking the form of safeguard classifiers that predict the harmfulness of a user prompt or an LLM output (Inan et al., 2023; Sharma et al., 2025; Mazeika et al., 2024; Markov et al., 2023). Recent work (Li et al., 2025) proposes online monitors that are designed for unfolding output, the setting we consider here. Such oversight models can provide strong signals that can be leveraged within a statistical framework, as we discuss in this work.

Statistical Monitoring Frameworks Detecting when a model degrades at inference time or exceeds a pre-defined risk level has traditionally been studied in the context of distribution shifts. Some monitoring approaches rely on confidence sequences (Howard et al., 2021; Shin et al., 2023) to test whether the risk remains acceptable at all deployment time steps. This has been evaluated both with (Podkopaev & Ramdas, 2021) and without (Amoukou et al., 2024) labels, as well as under model adaptation (Schirmer et al., 2025; Bar et al., 2024). Relatedly, e-processes (Ramdas et al., 2023; Ramdas & Wang, 2025) have been used to track evidence of risk violations over time (Timans et al., 2025; Prinster et al., 2025). In the context of LLMs, conformal prediction has been used to provide statistical trustworthiness, albeit in a classic offline setting (Cherian et al., 2024; Mohri & Hashimoto, 2024; Quach et al., 2023; Gui et al., 2024). A related line of work on optimising reasoning budget develops dynamic statistical frameworks that yield optimal stopping rules for exiting thinking mode; however, their primary goal is efficiency (Wang et al., 2026; You et al., 2025; Zeng et al., 2025; Wu et al., 2025; Jazbec et al., 2024). Another line of work (Davidov et al., 2025; Feldman & Romano, 2026) uses conformal survival analysis to construct PAC-type bounds on the time-to-unsafe-sampling of a given prompt. Most closely related to our work are agentic oversight methods (Wang et al., 2025), particularly Sadhuka et al. (2025), which tackles the same online monitoring setting.

B. Missed Detection Risk

We complement the false alarm risk of § 3.3 with its counterpart, the missed detection risk. The two risks trade off against each other: lowering one comes at the cost of raising the other. Importantly, risk control is particularly well suited to navigating this trade-off, as it lets the practitioner choose which risk to control and at what level. This stands in contrast to sequential hypothesis tests such as e-valuators (Sadhuka et al., 2025), where one can only control the false alarm rate without any guarantee on power. We first formalize the missed detection risk and describe how the calibration procedures of § 3.4 extend to it, and then present experiments on mathematical reasoning.

Missed Detection Risk. The probability that the monitor never raises an alarm given that the underlying sequence is unsafe is denoted

$$\mathcal{R}^{II}(\lambda) = \mathbb{P}(\forall\, t \geq 1 : s_t \geq \lambda \mid y = 0). \tag{7}$$

This is an expectation of the binary loss $\ell(s_{1:t}, \lambda) = \mathbf{1}\{\forall\, t \geq 1 : s_t \geq \lambda\}$ taken over unsafe samples only ($y = 0$), so $\mathcal{R}^{II}(\lambda) \in [0, 1]$; equivalently, $\mathcal{R}^{II}(\lambda) = 1 - \text{power}(\lambda)$. In contrast to $\mathcal{R}^{I}$, $\mathcal{R}^{II}(\lambda)$ is *monotonically decreasing* in λ: raising the threshold makes the monitor more eager to flag, reducing missed detections at the cost of additional false alarms.

Controlling the Missed Detection Risk. The two calibration procedures of § 3.4 extend to $\mathcal{R}^{II}$ with a single modification. Because the empirical risk is now monotonically *decreasing* in λ, the target constraint is satisfied for all sufficiently large thresholds, and we therefore select the *smallest* valid λ rather than the largest. This yields the most permissive monitor that still meets the desired guarantee. The empirical risk $\hat{\mathcal{R}}^{II}(\lambda; \mathcal{D}_{\text{cal}})$ is evaluated on the unsafe subset of the calibration data, $\{(x^{(i)}, o^{(i)}_{1:T}, y^{(i)}) \in \mathcal{D}_{\text{cal}} : y^{(i)} = 0\}$, of effective size n_0. For control in expectation, the threshold is

$$\hat{\lambda}_{\text{CRC}} := \min\left\{\lambda \in \Lambda \,:\, \frac{n_0}{n_0+1}\,\hat{\mathcal{R}}^{II}(\lambda; \mathcal{D}_{\text{cal}}) + \frac{1}{n_0+1} \leq \epsilon\right\}, \tag{8}$$

which guarantees $\mathbb{E}_{\mathcal{D}_{\text{cal}}}[\mathcal{R}^{II}(\hat{\lambda}_{\text{CRC}})] \leq \epsilon$. For control with high probability, let $U(\lambda, n_0, \delta)$ be a $(1-\delta)$ upper confidence bound on $\hat{\mathcal{R}}^{II}(\lambda; \mathcal{D}_{\text{cal}})$. The threshold is then

$$\hat{\lambda}_{\text{UCB}} := \min\{\lambda \in \Lambda \,:\, U(\lambda, n_0, \delta) \leq \epsilon\}, \tag{9}$$

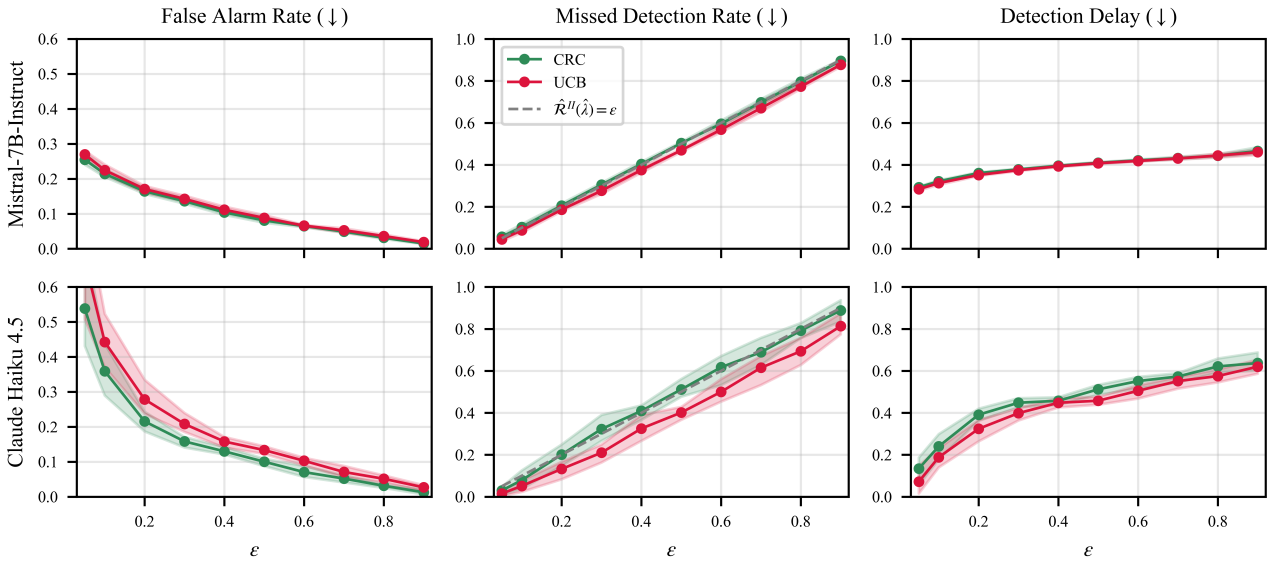

Figure 4. Monitoring performance when controlling missed detection risk $\mathcal{R}^{II}$ (Eq. (7)) instead of false alarm risk $\mathcal{R}^{I}$ (Eq. (2)): CRC and UCB control the risk $\mathcal{R}^{II}$ well (*second column*). E-valuators are excluded from this plot as they only allow to control the false alarm rate $\mathcal{R}^{I}$.

which guarantees $\mathbb{P}_{\mathcal{D}_{\text{cal}}}(\mathcal{R}^{II}(\hat{\lambda}_{\text{UCB}}) \leq \epsilon) \geq 1 - \delta$.

Controlling Missed Detection Risk on Factuality We now select the threshold to give a guarantee on the missed detection risk (one minus power) rather than on the false alarm risk. Keeping $\mathcal{R}^{II}$ low ensures a certain power level, at the cost of admitting more false alarms. We carry out this calibration in the mathematical reasoning setting of § 4.1. Notably, to our knowledge no existing LLM monitoring tool provides guarantees on the missed detection rate—including the e-valuator framework—meaning no direct baseline exists for this setting. Fig. 4 displays the results. The missed detection risk remains controlled at the prescribed level. At low missed detection risk, the false alarm rate is correspondingly high. For Claude—which produces a larger fraction of safe samples—the false alarm rate is higher, while for Mistral—which produces more incorrect samples—the false alarm rate is lower.

C. The E-valuator framework

We briefly summarize the *E-valuator* framework from Sadhuka et al. (2025) below. Given a prompt, the LLM produces a variable-length trajectory $o_{1:T}$, and after each step t a verifier is employed to return the score $s_t = p_\psi(y \mid x, o_{1:t})$ based on the (partial) trajectory thus far. The full score sequence $s_{1:T} = (s_1, \ldots, s_T)$ is paired with a final label $y \in \{0, 1\}$ indicating whether the final output is correct. In Sadhuka et al. (2025), the monitoring task is then cast as a *sequential hypothesis test* that distinguishes successful trajectories from unsuccessful ones via the class-conditional distributions of verifier scores:

$$H_0 : s_{1:T} \sim P_1 \qquad \text{vs.} \qquad H_1 : s_{1:T} \sim P_0,$$

where P_1 has density $p_1(s_{1:T}) = p(s_{1:T} \mid y = 1)$, i.e. correct, and P_0 has density $p_0(s_{1:T}) = p(s_{1:T} \mid y = 0)$, i.e. incorrect. For a new LLM trajectory, the procedure evaluates a statistic M_t online, and rejects the null once M_t crosses a threshold c_α. The target guarantee is *anytime-valid* false-alarm control, or

$$\mathbb{P}_{H_0}\big(\exists\, t \geq 1 : M_t \geq c_\alpha\big) \leq \alpha,$$

meaning that a true successful trajectory is incorrectly flagged at any time with probability at most α, for any unknown sequence length T.

E-valuator-anytime. The key mechanism is an *e-process* instantiated by the sequential density ratio $M_t = p_0(s_{1:t})/p_1(s_{1:t})$, which forms a test martingale under H_0 and therefore admits the threshold $c_\alpha = 1/\alpha$ with strict

time-uniform error control via Ville's inequality (Ramdas & Wang, 2025). Intuitively, M_t grows when the observed score history looks more consistent with the distribution of unsuccessful trajectories than with that of successful ones. Since $p_0(s_{1:t})$ and $p_1(s_{1:t})$ are unknown in practice, Sadhuka et al. (2025) estimate their ratio discriminatively using Bayes' rule. Accordingly, for each step t a probabilistic classifier $\hat{f}_t(s_{1:t}) \approx p(y = 1 \mid s_{1:t})$ is trained on calibration data, yielding a plug-in statistic that approximates the true likelihood ratio as

$$\hat{M}_t = \frac{1 - \hat{f}_t(s_{1:t})}{\hat{f}_t(s_{1:t})} \frac{\hat{\pi}_1}{1 - \hat{\pi}_1},$$

where $\pi_1 \approx p(y = 1)$ forms a base rate.

E-valuator-PAC. To empirically improve power, Sadhuka et al. (2025) also suggest a data-driven relaxation of the rejection threshold $1/\alpha$, which can be conservative in finite-horizon deployments. Leveraging an additional held-out calibration split of successful trajectories, they estimate the null distribution of the running maximum $\max_t \hat{M}_t$, and subsequently set c_α to a high-probability upper bound on its $(1 - \alpha)$-quantile. This yields a *probably-approximately-correct*-type (PAC) threshold with guarantee

$$\mathbb{P}_{D_{\text{cal}}} \left(\mathbb{P}_{H_0} \left(\exists\, t \geq 1 : \hat{M}_t \geq c_\alpha \mid D_{\text{cal}} \right) \leq \alpha \right) \geq 1 - \delta,$$

which weakens the exact anytime-valid guarantee to a high-probability guarantee over the calibration sample. Unlike the exact threshold, false alarm then need not hold for every realized calibration draw, but the alarm trigger can be substantially less conservative and therefore more powerful in practice.