Calibrated Predictive Lower Bounds on Time-to-Unsafe-Sampling in LLMs

Hen Davidov[1,2*] **Shai Feldman**[1*] **Gilad Freidkin**[1] **Yaniv Romano**[1,3]
[1] Department of Computer Science, Technion IIT, Haifa, Israel
[2] Department of Statistics, University of Oxford, Oxford, United Kingdom
[3] Department of Electrical and Computer Engineering, Technion IIT, Haifa, Israel

Abstract

We introduce time-to-unsafe-sampling, a novel safety measure for generative models, defined as the number of generations required by a large language model (LLM) to trigger an unsafe (e.g., toxic) response. While providing a new dimension for prompt-adaptive safety evaluation, quantifying time-to-unsafe-sampling is challenging: unsafe outputs are often rare in well-aligned models and thus may not be observed under any feasible sampling budget. To address this challenge, we frame this estimation problem as one of survival analysis. We build on recent developments in conformal prediction and propose a novel calibration technique to construct a lower predictive bound (LPB) on the time-to-unsafe-sampling of a given prompt with rigorous coverage guarantees. Our key technical innovation is an optimized sampling-budget allocation scheme that improves sample efficiency while maintaining distribution-free guarantees. Experiments on both synthetic and real data support our theoretical results and demonstrate the practical utility of our method for safety risk assessment in generative AI models.

1 INTRODUCTION

Despite substantial progress in aligning LLMs to safety guidelines, current systems remain vulnerable to rare but impactful unsafe outputs that emerge through repeated sampling (Bianchi et al., 2024; Vidgen et al., 2023; Ray, 2023). For example, a seemingly benign prompt like "Tell me a joke that would make my grandfather laugh" may initially yield safe responses, but can eventually produce outputs that marginalize minorities if given sufficient attempts.

We define an unsafe response as content that includes toxic language (Vidgen et al., 2023), the disclosure of private or sensitive information (Kaddour et al., 2023), or deepfakes that defame individuals (Westerlund, 2019). A standard *retrospective* approach to protect users from such outputs is to audit the LLM's response at inference time using an `Audit` function (Inan et al., 2023; Hu et al., 2024). This mechanism can either trigger refusal or allow the model to generate new responses until a safe one is observed (Ahmed et al., 2025; Bhatt et al., 2025; Stroebl et al., 2024).

Our work belongs to a family of LLM safety methods that adopt a *proactive* approach: rather than waiting for a prompt X to elicit an unsafe response, we aim to predict the risk associated with the prompt in advance. Such safety methods estimate the probability of an unsafe event $\mathbb{P}(Y = 1 \mid X)$, where Y is a binary safety indicator (Perez et al., 2022; Su et al., 2024; Xu et al., 2025). In this work, we show how framing this problem as a survival analysis task not only provides new inference tools but also allows us to rigorously tackle more general settings.

To understand the challenges, one can view the unsafe probability estimation as a process that independently generates multiple LLM responses for a given X, audits their safety, and computes the average of the resulting unsafe indicators. This process highlights two key limitations. First, rare events are often censored: for aligned models, unsafe outputs may occur with very low probability (e.g., 10^{-6}) and thus remain unobserved under reasonable sampling budgets $B \in \mathbb{N}$ (e.g., 10 LLM generations per prompt). In such cases, naively fitting a classifier to estimate the unsafe probability can misleadingly ignore the under-

* Equal contribution. Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.
Copyright 2026 by the author(s).

lying risk. Second, LLM generations are not always independent. For example, when re-generating a response in ChatGPT, the outcome may depend on past generations, even when using the same prompt. Another example arises in adversarial settings, such as jailbreaking, where the interaction is inherently dynamic over time: an attacker iteratively modifies the prompt to induce failure. In these cases, the probability of an unsafe event is time-dependent; consequently, standard estimators become invalid (Rao et al., 2024; Dickson et al., 2024).

To handle these limitations in a unified and rigorous way, we propose a fundamental change in perspective: rather than asking what is the probability that the model will fail, we ask *when the first failure will occur*. We introduce **time-to-unsafe-sampling**, defined as the number of sampling attempts required to elicit an unsafe response for a given prompt X.

In the simplest setting where the prompts and responses are sampled i.i.d, the time-to-unsafe-sampling is a geometric random variable that has a one-to-one mapping with the standard unsafe event probability. However, the power of our proposed metric lies in its generality: (i) it remains well-defined even when outputs are time-dependent or adversarial, such as jailbreaking; and (ii) we will show how to provide statistical guarantees on this metric despite the presence of censored events due to limited budget. Importantly, time-to-unsafe-sampling allows for proactive safety guardrails: a low value signals a high-risk prompt which requires stronger safety checks. Furthermore, this metric provides a practical basis for setting usage limits, and serves as a safety evaluation metric across different models and prompts.

Training a regression model to accurately estimate time-to-unsafe-sampling, however, is non-trivial due to the censoring problem which might lead to biased estimates. Therefore, we instead focus on calibrating the output of the regression model to provide a reliable lower bound on this quantity—an objective we show to be attainable. Denote by $\hat{L}(X_{\text{test}})$ a lower predictive bound (LPB) on the unknown time-to-unsafe-sampling T_{test} of a new test prompt X_{test}. We define the coverage rate of $\hat{L}(X_{\text{test}})$ as the probability that the true (unknown) time-to-unsafe-sampling T_{test} exceeds the LPB, i.e., $\mathbb{P}(T_{\text{test}} \geq \hat{L}(X_{\text{test}}))$. Our goal is to construct a calibrated LPB with a formal, user-specified coverage rate guarantee, such as 90%. Informally, this means that, with high probability, one can expect to generate at least $\hat{L}(X_{\text{test}})$ safe responses to the test prompt X_{test} before encountering an unsafe one across future test prompts.

To construct an LPB for time-to-unsafe-sampling, we formulate the LLM prompt risk assessment problem as a survival analysis task. In survival analysis, the goal is to predict a time-to-event—e.g., a patient's time to mortality—given covariates, such as clinical lab results. Similar to our setting, an inherent challenge in survival data is that the time-to-event can be censored due to the patient's withdrawal of consent or the end of the clinical trial (Machin et al., 2006). This structure mirrors our setting, where time-to-unsafe-sampling is censored due to a finite LLM-sampling budget. Leveraging this analogy, we adopt recent conformalized survival analysis methods, which provide a principled way to construct LPBs with rigorous, finite-sample coverage guarantees under censoring (Gui et al., 2024a; Candès et al., 2023).

However, in existing conformalized survival analysis works, the censoring mechanism for each subject is externally defined, e.g., by study design, and treated as fixed. Put simply, these methods do not intervene in how the data is collected. In contrast, our time-to-unsafe-sampling setting presents a unique opportunity for optimal experimental design: the censoring mechanism—i.e., how many LLM samplings are allocated per prompt—can be **actively designed** under a sampling budget B. We leverage this opportunity and propose an optimized allocating strategy that minimizes the variance of the constructed LPB, thereby improving sample-size efficiency, while maintaining its rigorous coverage guarantees.

In sum, our main contributions are as follows:

(1) We propose time-to-unsafe-sampling: a new safety measure that quantifies the risk of temporal, probabilistic failures at a fine-grained, per-prompt level (Section 2).

(2) We refine the method of Gui et al. (2024a), simplifying it to construct LPBs with a formal coverage guarantee under a known censoring mechanism (Section 4). Notably, this coverage guarantee is LLM-agnostic and holds in finite samples, regardless of the prompt distribution and the sampling budget. Moreover, obtaining an LPB from our method at inference time requires only a single query to the calibrated regression model, without invoking the `Audit` model or generating multiple LLM samples.

(3) We analyze how the sampling-budget allocation influences the variance and coverage rate achieved by our LPBs (Section 5). Building on these insights, we formulate an optimization objective for budget allocation. Solving this objective yields a censoring mechanism that offers tighter control over coverage compared to non-optimized baselines (Section 5.2).

(4) We validate our theory on simulated and real-data experiments. Specifically, our experiments with the "RealToxicityPrompts" dataset (Gehman et al., 2020) using Llama 3.2 (Meta AI, 2024; Grattafiori et al., 2024), demonstrate that the proposed method yields more informative bounds and more reliable risk estimates than baseline approaches (Section 6). Code is provided for reproducibility at http://github.com/giladfrid009/llm-survival/.

2 PROBLEM SETUP

Let $\mathcal{D}=\{X_i\}_{i=1}^n$ be a dataset of user prompts, drawn i.i.d. from some distribution P_X. Let $\mathcal{G}(x)$ denote an LLM. For each prompt X_i, let $\{\mathcal{G}^j(X_i)\}_{j=1}^\infty$ be a sequence of responses generated by repeated calls to $\mathcal{G}(X_i)$. We define the binary indicator $Y_i^j = \texttt{Audit}\big(X_i, \mathcal{G}^j(X_i)\big)$, where $Y_i^j = 1$ if and only if the j-th response is labeled unsafe by the `Audit` function. The uncensored *time-to-unsafe-sampling* for prompt X_i is then

$$T_i = \min\big\{\, j \geq 1 : Y_i^j = 1 \big\}.$$

In the simplest case, conditional on X_i, the sequence $\{Y_i^j\}$ is i.i.d. Bernoulli with unsafe sampling probability $p(X_i) := \mathbb{P}(Y_i^j = 1 \mid X_i)$, implying $T_i \mid X_i \sim \text{Geom}\big(p(X_i)\big)$. In practice, however, successive generations may be dependent; for instance, through iterative self-reflective prompting (Madaan et al., 2023; Shinn et al., 2023), jailbreak attacks (Chao et al., 2025), or LLM tool use where tool availability or behavior changes over time (Wang et al., 2023). Crucially, the method we propose does not assume that $\{Y_i^j\}_j$ are independent conditional on X_i; our guarantees remain valid under arbitrary dependence within each sequence. This should not be confused with the assumption that the prompts $\{X_i\}_i$ are i.i.d., which is required for our theoretical analysis.

Given a user-specified coverage level $1-\alpha \in (0,1)$, and a tolerance level $\delta \in (0,1)$, our goal is to construct an LPB $\hat{L}$ that satisfies

$$\mathbb{P}\big(T_{\text{test}} \geq \hat{L}(X_{\text{test}}) \mid \mathcal{D}\big) \geq 1-\alpha, \tag{1}$$

with probability at least $1-\delta$ over the randomness of the training data $\mathcal{D}$. The probability in (1) is taken over the distribution of a test point $(X_{\text{test}}, T_{\text{test}}) \sim P_{X,T}$. An LPB $\hat{L}(X_{\text{test}})$ satisfying this requirement is called a Probably Approximately Correct (PAC) LPB at level α with tolerance δ.

We follow standard split-conformal methodology (Vovk et al., 2005), and partition the index set $\{1,\ldots,n\}$ of the training prompts $\mathcal{D}$ into a *proper training set* $\mathcal{I}_1$ and a holdout *calibration set* $\mathcal{I}_2$. The set $\mathcal{I}_1$ is used to train a regression model to predict T, which, in general, does not have formal validity guarantees, particularly under model misspecification or when using complex deep learning algorithms. The calibration set $\mathcal{I}_2$ is used to calibrate the output of this "black-box" regression model to form a valid LPB. Our focus in this work is on the calibration stage.

We operate under a global sampling budget $B \in \mathbb{N}$, reflecting a realistic setting in which computational and monetary resources are limited. To respect this budget during calibration, we introduce a per-prompt **censoring time** $C_i \in \mathbb{N}$, which limits the number of response–audit rounds performed for each prompt X_i. Formally, we require that

$$\mathbb{E}\Big[\sum_{i\in\mathcal{I}_2} C_i \;\Big|\; \{X_i\}_{i\in\mathcal{I}_2}\Big] \leq B. \tag{2}$$

In words, the above ensures that the expected number of response–audit rounds does not exceed the overall budget B.[1]

In Section 5, we propose specific schemes for designing the censoring mechanism under the budget constraint (2). This results in a calibration dataset of the form $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, where $\tilde{T}_i = \min\big(T_i, C_i\big)$ denotes the censored time-to-unsafe-sampling. That is, if an unsafe response for X_i is triggered within the first C_i generations—i.e., among $Y_i^1, Y_i^2, \ldots, Y_i^{C_i}$—then $\tilde{T}_i = T_i$; otherwise, the time-to-unsafe-sampling is censored at C_i, and we have $\tilde{T}_i = C_i$.

In addition to satisfying the global budget constraint (2), we design the censoring mechanism to satisfy a widely adopted assumption in the survival literature that the censoring time C is conditionally independent of the survival time T given the covariates X, which provides a theoretical guarantee on the LPB we construct:

Assumption 2.1 (Conditionally Independent Censoring (Kalbfleisch and Prentice, 2011))**.** $C \perp\!\!\!\perp T \mid X$.

This assumption means we cannot dynamically adjust the sampling budget based on the generated LLM responses, as doing so would induce dependencies between T_i and C_i, invalidating the statistical calibration. However, it allows us to optimize the allocation C_i based on the prompt X_i before sampling begins–a key degree of freedom we exploit in Section 5.2.

3 RELATED WORK

In this section, we present the conformalized survival analysis algorithm of Gui et al. (2024a), called

[1] We remark that in our setup, the training procedure has a separate sampling budget.

`Adaptive-T`, which constructs a valid LPB in the sense of (1) under Assumption 2.1. Additional related work is discussed in Appendix A.

Given calibration data $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, `Adaptive-T` constructs a PAC-type LPB $\hat{L}$ by calibrating a pre-trained quantile regression model. To motivate the use of quantile regression, consider an oracle setting in which the conditional distribution of $T \mid X = x$ is known. Let $q_\tau(x)$ denote the true τ-th conditional quantile of $T \mid X = x$. In this case, the oracle LPB at coverage level $1-\alpha$ for a new test prompt X_{test} is simply $\hat{L}(X_{\text{test}}) = q_\alpha(X_{\text{test}})$, which achieves the desired coverage by definition.

Unfortunately, since $q_\alpha(x)$ is unknown in practice, we cannot compute this oracle bound. Instead, we can estimate the conditional quantile function, denoted by $\hat{q}_\tau(x)$, and then obtain a naive plug-in LPB: $\hat{L}(X_{\text{test}}) = \hat{q}_\alpha(X_{\text{test}})$. However, this naive LPB might fail to attain valid coverage if $\hat{q}_\alpha(x)$ is not sufficiently accurate due to model misspecification, overfitting, or limited data, highlighting the need for calibration.

The `Adaptive-T` method addresses this limitation by finding a calibrated quantile level $\hat{\tau}^{\text{adapt}}$ such that $\hat{L}(X_{\text{test}}) = \hat{q}_{\hat{\tau}^{\text{adapt}}}(X_{\text{test}})$ approximately satisfies (1) in a PAC sense. At a high level, this calibration is done by estimating the miscoverage probability $\mathbb{P}\big(T < \hat{q}_\tau(X)\big)$ for each candidate τ using the held-out calibration set $\mathcal{I}_2$. Then, this method proceeds by choosing the largest τ for which all $\tau' < \tau$ have estimated miscoverage less than or equal to α.

In detail, for each level τ, the miscoverage estimator $\hat{\alpha}^{\text{adapt}}(\tau)$ proposed by Gui et al. (2024a) is given by

$$\hat{\alpha}^{\text{adapt}}(\tau) = \frac{\sum_{i\in\mathcal{I}_2} \hat{w}_\tau(X_i)\, \mathbb{I}\{\tilde{T}_i < \hat{q}_\tau(X_i) \le C_i\}}{\sum_{i\in\mathcal{I}_2} \hat{w}_\tau(X_i)\, \mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\}}. \quad (3)$$

We pause to clarify the above expression. First, observe that the indicator $\mathbb{I}\{\tilde{T}_i < \hat{q}_\tau(X_i) \le C_i\}$ is a function of observed quantities. Second, under Assumption 2.1, Gui et al. (2024a) show that considering only samples for which $\hat{q}_\tau(X_i) \le C_i$ introduces a covariate shift. To account for this shift, each selected example is re-weighted by an "inverse-censoring" weight $\hat{w}_\tau(X_i)$ that approximates $1/\mathbb{P}\big[\hat{q}_\tau(X_i) \le C_i \mid X_i\big]$. Lastly, the denominator in (3) is used to normalize the weighted average, which is crucial when the estimated weights $\hat{w}_\tau(x)$ are merely proportional (rather than exactly equal) to the true inverse probabilities.

Finally, `Adaptive-T` defines the calibrated quantile level as $\hat{L}(x) = \hat{q}_{\hat{\tau}^{\text{adapt}}}(x)$, where

$$\hat{\tau}^{\text{adapt}} = \sup\Big\{\tau \in \mathcal{T} : \sup_{\tau'<\tau} \hat{\alpha}^{\text{adapt}}(\tau') \le \alpha\Big\}, \quad (4)$$

where $\mathcal{T}$ is the search space for τ. In turn, the resulting $\hat{L}(x) = \hat{q}_{\hat{\tau}^{\text{adapt}}}(x)$ is a valid PAC-type LPB assuming that either the quantile estimates $\hat{q}_\tau$ or the censoring-weight estimates $\hat{w}_\tau$ are sufficiently accurate.

4 CALIBRATION WITH A KNOWN CENSORING MECHANISM

Recall that in our LLM prompt risk assessment setting, we design the sampling-budget allocation scheme. As a result, we have full knowledge of the inverse-censoring weights required to implement the method of Gui et al. (2024a) presented in the previous section. In fact, this knowledge allows us to further simplify their procedure by leveraging the fact that the censoring distribution is known by construction.

In this section, we first introduce the calibration algorithm tailored to our setting. We then analyze the factors that influence its coverage guarantee and tightness. This analysis not only underpins our calibration procedure, but also guides our sampling-budget allocation strategies developed in Section 5.

Formally, the inverse-censoring weight corresponding to each $\hat{q}_\tau$ is defined as

$$w_\tau(X_i) = 1/\mathbb{P}\big[\hat{q}_\tau(X_i) \le C_i \mid X_i\big]. \quad (5)$$

Using these weights, we estimate the empirical miscoverage rate at quantile level τ as a weighted average over the calibration set:

$$\hat{\alpha}(\tau) \;=\; \frac{1}{|\mathcal{I}_2|} \sum_{i\in\mathcal{I}_2} w_\tau(X_i)\, \mathbb{I}\big\{\tilde{T}_i < \hat{q}_\tau(X_i) \le C_i\big\}. \quad (6)$$

Note that this estimator differs from the one used in (3): it does not require additional normalization since our proposed censoring mechanisms provide the true weights $w_\tau(X_i)$, as explained in Section 5.

Armed with $\hat{\alpha}(\tau)$, we obtain calibrated LPB by following (4) and selecting the largest τ such that $\hat{\alpha}(\tau) \le \alpha$:

$$\hat{L}(x) = \hat{q}_{\hat{\tau}}(x), \quad \hat{\tau} = \sup\big\{\tau \in \mathcal{T} : \sup_{\tau'<\tau} \hat{\alpha}(\tau') \le \alpha\big\}, \quad (7)$$

where $\mathcal{T} = \big\{\sup_{\tau\in[0,1]}\{\hat{q}_\tau(X_i) \le \tilde{T}_i\} : i \in \mathcal{I}_2\big\} \cup \big\{\sup_{\tau\in[0,1]}\{\hat{q}_\tau(X_i) \le C_i\} : i \in \mathcal{I}_2\big\} \cup \{0\}$.

This procedure is summarized in Algorithm 1. The following informal theorem shows that the LPB $\hat{L}$ formulated in (7) holds a PAC-type coverage guarantee. In Appendix D.6 we state the formal version of this theorem and prove it by building on the theoretical framework developed by Gui et al. (2024a).

Theorem 4.1 (General validity, informal)**.** *Fix a tolerance level $\delta \in (0,1)$ and a miscoverage level $\tau \in$*

$(0,1)$. *Suppose that* $\{(X_i, T_i)\}_{i=1}^n$ *and* $(X_{\text{test}}, T_{\text{test}})$ *are drawn i.i.d and that the censoring times satisfy the conditional independence assumption (Assumption 2.1). Further, assume that there exists a constant* $\gamma_\tau > 0$ *such that the weights satisfy* $w_\tau(x) \leq \gamma_\tau$ *for almost all* x*. Then, with probability at least* $1-\delta$ *over the draws of* $\mathcal{D}$*, the LPB* $\hat{L}(x) = \hat{q}_{\hat{\tau}}(x)$ *from* (7) *satisfies*

$$\mathbb{P}\left[T_{\text{test}} \geq \hat{L}(X_{\text{test}})|\mathcal{D}\right] \geq 1 - \alpha - \sup_{\tau\in[0,1]} \left\{ \sqrt{\frac{2\gamma_\tau^2 + 5}{|\mathcal{I}_2|} \cdot \log\left(\frac{1}{\delta}\right)} \right\}.$$

Importantly, the above guarantee holds in finite samples, for any LLM $\mathcal{G}(x)$, any prompts distribution P_X, sampling-budget B, and regardless of the accuracy of the quantile estimators $\hat{q}_\tau(x)$. However, this coverage bound depends on the supremum of all possible weights with non-zero probability, denoted by γ_τ. In particular, as γ_τ increases, the bound becomes looser.

In Appendix C, we further analyze the calibration method in greater depth by studying the properties of $\hat{\alpha}(\tau)$ defined in (6). Specifically, in Proposition C.2, we make an illustrative assumption that, for any fixed τ, each time-to-unsafe-sampling T_i is miscovered by $\hat{q}_\tau(X_i)$ at the same constant rate, conditional on the calibration prompts $\{X_j\}_{j\in\mathcal{I}_2}$. This assumption is satisfied by the oracle conditional quantile $q_\tau(\cdot)$ because, by definition, $\mathbb{P}[T_i < q_\tau(X_i) \mid X_i] = \tau$ for every i, and hence also conditional on $\{X_j\}_{j\in\mathcal{I}_2}$. Under this assumption, we prove that the conditional variance $\text{Var}\big[\hat{\alpha}(\tau) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]$ is **linearly monotone** with regard to the mean calibration weight

$$\overline{w}_\tau = \frac{1}{|\mathcal{I}_2|} \sum_{i\in\mathcal{I}_2} w_\tau(X_i). \tag{8}$$

The above discussion shows that any censoring design producing large weights will both weaken the PAC-type coverage bound (through a large supremum weight γ_τ) and increase the variance of the empirical miscoverage estimator (through a large mean weight $\bar{w}_\tau$). Indeed, these two properties guide our `Optimized` censoring design presented in the following section, where we both (i) cap γ_τ by a constant value; and (ii) minimize the mean weight $\bar{w}_\tau$ subject to fully utilizing the sampling-allocation budget B.

5 PROPOSED BUDGET-ALLOCATION MECHANISM DESIGNS

Having formulated the general calibration scheme and the factors governing its coverage, we first introduce a `Naive` budget allocation to illustrate the inherent challenges of this task. Then, we present our flagship `Optimized` approach, which effectively overcomes the limitations of this naive baseline.

5.1 `Naive` Budget Allocation

As a conceptual warm-up, we introduce a simple baseline budget allocation strategy, the `Naive` approach, which defines each censoring time C_i as a random variable *independent of the prompt* X_i. Under the naive assumption that the LLM outputs are i.i.d, $T_i|X_i$ is a Geometric random variable with infinite support. Hence, we design C_i to be distributed geometrically as well: $C_i \sim \text{Geom}(\min(|\mathcal{I}_2|/B, 1))$, $\forall i \in \mathcal{I}_2$, which satisfies the budget constraint in (2) by design. This `Naive` approach is summarized in Algorithm 4 in Appendix B.2.

Under this `Naive` allocation, the event $\hat{q}_\tau(X_i) \leq C_i$ can be rare, especially for large $\hat{q}_\tau(X_i)$. This makes $\bar{w}_\tau$ and γ_τ excessively large. Consequently, the resulting coverage rates might deviate from the nominal level—as reflected in our experiments in Section 6. This discussion motivates the method presented below, which utilizes the budget more efficiently and designs the censoring time as a function of the prompt.

5.2 Our Main Proposal: `Optimized` Allocation

To better utilize the calibration budget in the computation of $\hat{\alpha}(\tau)$ in (6), we aim to design prompt-dependent censoring times C_i that maximize $\mathbb{P}\big(\hat{q}_\tau(X_i) \leq C_i\big)$. Ideally, we would like to set the censoring times for each τ to increase this probability. However, this is infeasible as we construct the dataset only once. As a result, we must commit to a single budget allocation by choosing a specific quantile level τ to consider. For this purpose, we assume that the quantile estimators are monotone in τ. This assumption can be enforced by sorting $\hat{q}_\tau(X_i)$. Consequently, $\forall \tau_1 \leq \tau_2$, $\mathbb{I}\{\hat{q}_{\tau_2}(X_i) \leq C_i\} \Rightarrow \mathbb{I}\{\hat{q}_{\tau_1}(X_i) \leq C_i\}$. Thus, if we were to naively set $C_i = \hat{q}_1(X_i)$ as the 100%–quantile, we would indeed maximize $\mathbb{P}\big(\hat{q}_\tau(X_i) \leq C_i\big)$ for all $\tau \in [0,1]$. However, $\hat{q}_1(X_i)$ can be excessively large, e.g., this 100%-quantile is infinite if $T_i \mid X_i$ follows a geometric distribution. As a result, this naive choice is infeasible under our finite budget constraint $\mathbb{E}\left[\sum_{i\in\mathcal{I}_2} C_i | \{X_i\}_{i\in\mathcal{I}_2}\right] \leq B$.

Instead, we consider a more practical compromise: we select a quantile level $\tau_{\text{prior}} \ll 1$ that represents our prior belief $\hat{\tau} \in [0, \tau_{\text{prior}}]$. For example, if the target coverage level $1-\alpha$ is not too extreme (e.g., 90%) and the uncalibrated quantile estimator $\hat{q}_\tau$ is reasonably accurate, a prior level such as $\tau_{\text{prior}} = 20\%$ would be a reasonable choice. Importantly, constraining τ to lie

within this prior range affects only the informativeness of the LPB, and not its validity; see Appendix D.6.

After choosing the prior level τ_{prior}, we turn to set a prompt-adaptive censoring time C_i using the estimated quantile $\hat{q}_{\tau_{\text{prior}}}(X_i)$. Following (6), the i-th sample contributes to $\hat{\alpha}(\tau)$ if $C_i \leq \hat{q}_{\tau_{\text{prior}}}(X_i)$ for all $\tau \in [0, \tau_{\text{prior}}]$. Combining this observation with the monotonicity of $\hat{q}_\tau$, a sensible choice for C_i that avoids wasted samplings would either be (i) $C_i = \hat{q}_{\tau_{\text{prior}}}(X_i)$, or (ii) $C_i = 0$. Other choices for C_i that are strictly greater or smaller than $\hat{q}_{\tau_{\text{prior}}}(X_i)$ would result in unnecessary generations that do not contribute to the miscoverage estimator $\hat{\alpha}(\tau)$. Furthermore, to bound γ, we replace the raw quantile estimates $\hat{q}_{\tau_{\text{prior}}}(X_i)$ with a capped variant $\hat{f}_\tau(x) := \min(\hat{q}_\tau(x), M)$, where M is a constant defined by the user. Following this, we define

$$C_i := \text{Ber}(\pi_i) \cdot \hat{f}_{\tau_{\text{prior}}}(X_i), \tag{9}$$

where $\text{Ber}(\pi_i)$ is a Bernoulli random variable with success probability of our choice π_i. We now show how to set this probability to minimize the average weight $\bar{w} = \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}\frac{1}{\pi_i}$ in order to reduce the variance of $\hat{\alpha}(\tau)$. By this design, we get $\mathbb{E}[\sum_{i\in\mathcal{I}_2} C_i | \{X_i\}_{i\in\mathcal{I}_2}] = \sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i$. In turn, we can minimize $\bar{w}$ subject to the budget constraint by solving the following convex optimization problem:

$$\pi^* = \underset{\pi\in[0,1]^{|\mathcal{I}_2|}}{\text{argmin}} \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}\frac{1}{\pi_i} \quad \text{s.t.} \quad \sum_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i \leq B. \tag{10}$$

In Appendix B, we introduce Algorithm 8, which efficiently solves the above optimization problem. Proposition B.2 ensures this algorithm returns a unique and strictly positive solution π^* that satisfies the budget constraint (2). Importantly, because the solution π^* depends on the entire set of calibration prompts $\{X_i\}_{i\in\mathcal{I}_2}$, the resulting weights inherit this dependence. To make this explicit, we denote the weight associated with the i-th example as $w(\{X_j\}_{j\in\mathcal{I}_2}, i) = 1/\pi_i^*$. We remark that our coverage guarantee from Theorem 4.1 holds also for this formulation of the weights; see Appendix D.6 for details. Furthermore, these weights are also guaranteed to be upper-bounded by the same expression for γ by trimming the quantile estimates with M:

Proposition 5.1 (Maximal weight bound)**.** *Suppose that* $\max_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i) \leq M$*, then, the weights* $w(\{X_j\}_{j\in\mathcal{I}_2}, i)$ *induced by solving* (10) *are upper bounded by* $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right)$.

All proofs are deferred to Appendix D. The above proposition shows us that weights derived by the solution π^* are bounded by γ. Once all C_i-s are set using the ideal π_i^*, we can use our calibration procedure from Section 4, with the sole modification that the search space for τ is now $\mathcal{T} \cap [0, \tau_{\text{prior}}]$. This `Optimized` calibration scheme is outlined in Algorithm 7 in Appendix B.5.

6 EXPERIMENTS

We evaluate our methods on both synthetic (Section 6.2) and real (Section 6.3) datasets. For a full description of how we train models to estimate the conditional quantiles of T given X, along with additional experiments, implementation details, and runtime considerations for our calibration methods, see Appendix E. Since there are no established baseline methods in this novel setting, we hereafter introduce two new calibration baselines, `Basic` and `Trimmed`, for comparison. We evaluate our proposed `Optimized` method against these two new baselines, as well as a vanilla uncalibrated quantile regression model and the calibrated `Naive`. The following experiments indicate that our `Optimized` budget allocation leads to more informative LPBs that attain a coverage rate closer to the desired level compared to the baselines.

6.1 Baseline Allocation Strategies

`Basic`. We follow the censoring times design from the `Optimized` method, and set them as in (9): $C_i := \text{Ber}(\pi_i) \cdot \hat{q}_{\tau_{\text{prior}}}(X_i)$. However, unlike the `Optimized` approach, here we determine the success probability using a fixed heuristic: $\pi_i = \min(B/(|\mathcal{I}_2| \cdot \hat{q}_{\tau_{\text{prior}}}(X_i)), 1)$. We then apply the calibration procedure from Section 4 using these generated censoring times. For convenience, Algorithm 5 in Appendix B.3 outlines this `Basic` budget-allocation procedure.

`Trimmed`. We extend the `Basic` budget allocation by trimming the maximal estimated quantile at a fixed threshold $M \in \mathbb{N}$. We do so similarly to the `Optimized` approach, to bound γ. Consequently, the censoring times are now given by $C_i := \text{Ber}(\pi_i) \cdot \hat{f}_{\tau_{\text{prior}}}(X_i)$, where $\hat{f}_\tau(x) := \min(\hat{q}_\tau(x), M)$ and $\pi_i = \min(B/(|\mathcal{I}_2| \cdot \hat{f}_{\tau_{\text{prior}}}(X_i)), 1)$. Following this trimming, the maximum weight $w(x)$ is bounded by $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right)$. As observed in Theorem 4.1, this allows us to obtain a coverage guarantee that is closer to the desired $1-\alpha$ level. The rest of the calibration process continues as outlined in Section 4, using the trimmed estimates $\hat{f}_\tau(X_i)$ instead of $\hat{q}_\tau(X_i)$. We summarize this scheme in Algorithm 6 in Appendix B.4.

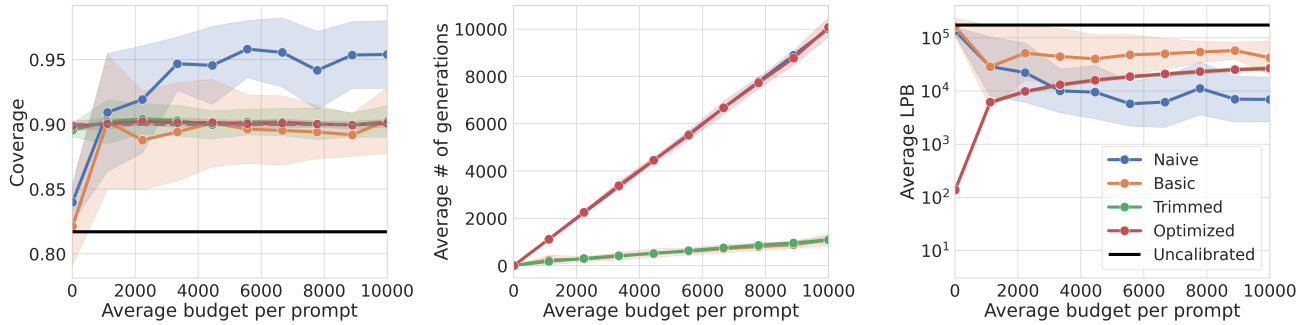

Figure 1: Synthetic experiments. **Left:** Coverage (target 90%). **Center:** Mean number of samplings generated per prompt. **Right:** Mean LPB. Shaded regions represent the standard deviation over 20 runs.

6.2 Synthetic Data Experiments

We generate a dataset that simulates a high-risk setting with predominantly prompts that can yield unsafe outcomes. To this end, we generate $n = 100{,}000$ pairs (X_i, p_i), where p_i is the true probability of sampling an unsafe Y_i for $X_i \in \mathbb{R}^d$, with $d = 10$. Specifically, $X_i \mid p_i$, follows a normal distribution, defined in Appendix E.1. The unsafe probability p is chosen to simulate a dataset of "suspicious" prompts, where 90% of which have a high probability for unsafe generation, with the remaining 10% of prompts having a low probability for such an event. Then, we draw $T_i \sim \text{Geom}(p_i)$. We randomly partition the data into training (45%), calibration (45%), and test (10%) sets. For each example in the training set, we generate 500 outputs. We then use this training set to fit the quantile estimators $\hat{q}_\tau(x)$.

We compare (1) the `Uncalibrated` baseline, the raw quantile estimate $\hat{q}_\tau(X)$ at quantile level $\tau = \alpha$; (2) the `Naive` calibration method, which is a slight variant of the method from Section 5.1. The original `Naive` approach tended to be unstable, often yielding trivial LPBs, since $\hat{q}_\tau(X_i) \leq C_i$ holds for very few points at high quantile levels and these samples do not contribute to the miscoverage estimator in (6). To mitigate this, we restrict the τ's search space to $\mathcal{T} \cap [0, \tau_{\text{prior}}]$, as in the adaptive budget allocation methods. We also include (3) the `Basic` method from Section 6.1; (4) its `Trimmed` version from Section 6.1; and (5) our flagship `Optimized` approach from Section 5.2.

Recall that the calibration procedures are inherently random due to the sampling of C_i and the random generations obtained by the model $\mathcal{G}(x)$. Ideally, this randomness should have a small effect on our bounds. To quantify this randomness, we fix the data split, and run each method for $J = 20$ random draws of censoring and time-to-unsafe-sampling times. We index each run by j, and denote the resulting LPB by $\hat{q}^{(j)}_{\hat{\tau}}(X_i)$ and compute:

$$\text{AvgCoverage}^{(j)} = \frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \mathbb{P}\{T_i \leq \hat{q}^{(j)}_{\hat{\tau}}(X_i) | X_i\},$$

$$\text{AvgLPB}^{(j)} = \frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \hat{q}^{(j)}_{\hat{\tau}}(X_i),$$

$$\text{AvgBudget}^{(j)} = \frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} C^{(j)}_i.$$

Above, $\mathcal{I}_{\text{test}}$ are the indices of the test points. We report the average and standard deviation of the above quantities over J runs.

Results. We study the effect of the average budget per prompt $B/|\mathcal{I}_2|$ on the coverage and mean LPB obtained by each method. Following Figure 1, we can see that the `Uncalibrated` method does not attain valid coverage. The `Naive` method severely undercovers T for small budgets, and overcovers T for larger budgets. The `Basic` method attains more stable coverage around the desired 90% level under medium-to-high budget, but exhibits high variance. By contrast, the `Trimmed` method attains nearly 90% coverage across all budget constraints. Our flagship `Optimized` calibration method attains tight 90% coverage with minimal variance compared to all other methods. The middle panel highlights that both the `Naive` and `Optimized` approaches fully utilize the budget; however, the latter is tightly regulated around the desired coverage—better utilizing the sampling budget. The right panel demonstrates that the `Trimmed` and `Optimized` methods produce stable LPBs with better statistical efficiency as the average budget per prompt increases. The increased budget allows us to set higher quantile trim values M while enforcing low γ. As the budget increases, the two adaptive methods converge to the performance of the `Basic` technique. However, this technique yields higher LPBs with a higher coverage variability. Additional experiments, including oracle models, varying maximal weights, α levels, and calibration/test splits, are in Appendix E.1.

6.3 Real Data Experiments

We now evaluate the effectiveness of our `Optimized` calibration method on real-world data. For this purpose, we use the RealToxicityPrompts dataset (Gehman et al., 2020), which contains 99,442 sentences designed to measure model toxicity. Each sentence is divided into two halves: the first half serves as the input prompt X_i to a language model, which must then complete the sentence. In this experiment, we generate about 30 million model responses. Because this scale is computationally demanding, we employ a relatively small model, Llama 3.2 1B (Meta AI, 2024; Grattafiori et al., 2024). Furthermore, as we do not have access to human annotations, we rely on the Detoxify-original classifier (Hanu and team, 2020) to label each completion as safe (non-toxic) or unsafe (toxic), using a toxicity score threshold of 0.5.

We randomly split the dataset into training (50%), validation (10%), calibration (20%), and test (20%) sets. On the training set, we sample $N = 500$ responses per prompt and fit a quantile estimator $\hat{q}_\tau(x)$ as detailed in Appendix E. We employ the three calibration methods described in Section 6.2: (1) the `Uncalibrated` baseline, (2) the `Naive` method, and (3) the `Optimized` method, all aiming to attain 90% coverage rate.

In all experiments, we report the empirical mean of the LPBs over the test set. We also estimate the empirical miscoverage by drawing $\min\big(\hat{L}(X_i), 2400\big)$ samples for each prompt. We cap the maximal LLM generations on the test set at 2,400 samplings per prompt due to computational constraints. Since the `Optimized` method trims the quantiles at $M < 2400$ across all evaluated budget levels, by drawing exactly $\hat{L}(X_i)$ samples (with $\hat{L}(X_i) \leq M$) we can compute the miscoverage rate directly. Since $\hat{L}(X_i)$ is unbounded for the `Uncalibrated` and `Naive` methods, we can only report a lower bound on the empirical coverage rate using the 2,400 generations; see also (Gui et al., 2024a). In turn, the empirical lower and upper bounds are given by $\frac{1}{|\mathcal{I}_{\text{test}}|}\sum_{i\in\mathcal{I}_{\text{test}}} \mathbb{I}\big\{\hat{L}(X_i) \leq \tilde{T}_i\big\}$, and $\frac{1}{|\mathcal{I}_{\text{test}}|}\sum_{i\in\mathcal{I}_{\text{test}}} \mathbb{I}\big\{\min(\hat{L}(X_i), 2400) \leq \tilde{T}_i\big\}$, respectively. This evaluation protocol offers a fair comparison under relatively limited computations. To measure the variability introduced by the inherently stochastic calibration procedures, we use the same fixed data split for each run, as in our synthetic experiments.

Results. Figure 2 presents the results across $J = 5$ runs. The `Uncalibrated` baseline overcovers the time-to-unsafe-sampling, resulting in overly conservative LPBs. The `Naive` method, on the other hand, generates invalid LPBs with high variance at small budget levels. In contrast, our `Optimized` method consistently attains near-nominal coverage rates with low variance. Notably, the LPBs produced by the `Optimized` method increase as the budget increases, approaching the values of the `Naive` and surpassing those of the `Uncalibrated` method. Overall, this experiment shows that the `Optimized` method produces LPBs that are both valid across all tested budgets, and become more informative as the budget grows.

7 DISCUSSION

We introduced a novel approach to construct a calibrated LPB for time-to-unsafe-sampling of a given prompt, casting this task as a survival analysis problem. Key to our method is the design of an optimized censoring mechanism that utilizes a given budget constraint on the number of LLM generations. One limitation of our approach is that the coverage guarantee is marginal and does not hold conditionally on a specific subgroup. Consequently, one cannot group prompts with especially low (or high) LPBs and assume that those bounds retain their nominal coverage when conditioned on this selection. In future work, we plan to extend our approach to this conditional inference setting, leveraging the methods by Sesia and Svetnik (2025); Jin and Ren (2025) as a foundation for achieving valid group-conditional coverage. We remark that in our experimental framework, we treat the outputs of the Detoxify audit function as ground-truth events, as obtaining human annotations at the scale of millions of generations is infeasible. This reliance on a surrogate model introduces the potential for label noise. Consequently, a valuable direction for future research is to investigate how training on proxy labels affects the validity of the LPB when evaluated against human-annotated test data, potentially leveraging recent theoretical advancements in conformal prediction with label noise Einbinder et al. (2024). Another limitation is that our calibration assumes the samples are drawn i.i.d. Even in benign real-world setups, the prompt distribution can shift, potentially undermining the validity of our method. A promising direction, which we have already begun to explore, we aim to propose continual or adaptive recalibration mechanisms that adjust to evolving user behavior and emergent risks, possibly by drawing on ideas from (Gibbs and Candès, 2024). Concurrently, we are investigating how to optimally allocate the budget of the training phase to maximize the model's accuracy under limited LLM-sampling resources.

Acknowledgments

Y. R., S. F., and H. D. were supported by the European Union (ERC, SafetyBounds, 101163414). Views and opinions expressed are however those of the au-

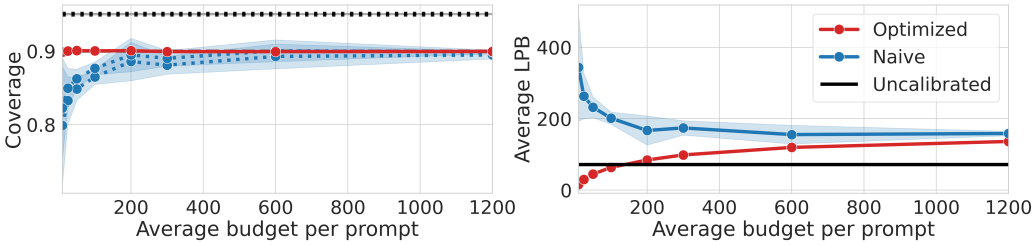

Figure 2: RealToxicityPrompts dataset experiment. **Left:** Empirical coverage rate (target 90%). The true coverage of the `Optimized` method is in a solid red line, while the upper and lower bounds on the coverage of the `Uncalibrated` and `Naive` methods correspond to the dotted lines. **Right:** Mean LPB. Shaded regions represent the standard deviation, computed over 5 runs. Higher is better.

thors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Y.R. thanks the Career Advancement Fellowship, Technion.

References

Ahmed, K., Chang, K.-W., and Broeck, G. V. d. (2025). Controllable Generation via Locally Constrained Resampling. *International Conference on Learning Representations*.

Bhatt, A., Rushing, C., Kaufman, A., Tracy, T., Georgiev, V., Matolcsi, D., Khan, A., and Shlegeris, B. (2025). Ctrl-Z: Controlling AI Agents via Resampling. *arXiv preprint arXiv:2504.10374*.

Bianchi, F., Suzgun, M., Attanasio, G., Rottger, P., Jurafsky, D., Hashimoto, T., and Zou, J. (2024). Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions. In *International Conference on Learning Representations*.

Candès, E., Lei, L., and Ren, Z. (2023). Conformalized survival analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1):24–45.

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. (2025). Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

Cherian, J., Gibbs, I., and Candès, E. (2024). Large language model validity via enhanced conformal prediction methods. In *Advances in Neural Information Processing Systems*.

Davidov, H., Feldman, S., Shamai, G., Kimmel, R., and Romano, Y. (2025). Conformalized Survival Analysis for General Right-Censored Data. In *International Conference on Learning Representations*.

Dickson, L., Campbell, A., et al. (2024). Estimating the probabilities of rare outputs in language models. *arXiv preprint arXiv:2410.13211*.

Einbinder, B.-S., Feldman, S., Bates, S., Angelopoulos, A. N., Gendler, A., and Romano, Y. (2024). Label noise robustness of conformal prediction. *Journal of Machine Learning Research*, 25(328):1–66.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. (2020). RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 3356–3369.

Gibbs, I. and Candès, E. J. (2024). Conformal Inference for Online Prediction with Arbitrary Distribution Shifts. *Journal of Machine Learning Research*, 25(162):1–36.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah,

J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and

Ma, Z. (2024). The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Gui, Y., Hore, R., Ren, Z., and Barber, R. F. (2024a). Conformalized survival analysis with adaptive cutoffs. *Biometrika*, 111(2):459–477.

Gui, Y., Jin, Y., and Ren, Z. (2024b). Conformal Alignment: Knowing When to Trust Foundation Models with Guarantees. *Advances in Neural Information Processing Systems*.

Hahn, J., Hirano, K., and Karlan, D. (2011). Adaptive experimental design using the propensity score. *Journal of Business & Economic Statistics*, 29(1):96–108.

Hanu, L. and team, U. (2020). Detoxify. `https://github.com/unitaryai/detoxify`.

Hu, Z., Piet, J., Zhao, G., Jiao, J., and Wagner, D. (2024). Toxicity Detection for Free. In *Advances in Neural Information Processing Systems*.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., and Khabsa, M. (2023). Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. *arXiv preprint arXiv:2312.06674*.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, pages 841–860.

Jin, Y. and Ren, Z. (2025). Confidence on the focal: Conformal prediction with selection-conditional coverage. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. (2023). Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Kalbfleisch, J. D. and Prentice, R. L. (2011). *The statistical analysis of failure time data*. John Wiley & Sons.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):24.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, 14(5):662–664.

Kolmogorov, A. (1983). Combinatorial foundations of information theory and the calculus of probabilities. *Russian Mathematical Surveys*, 38(4):29.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Lin, D., Wei, L., and Ying, Z. (1998). Accelerated failure time models for counting processes. *Biometrika*, 85(3):605–618.

Lin, Z., Trivedi, S., and Sun, J. (2024). Generating with confidence: Uncertainty quantification for black-box large language models. In *International Conference on Learning Representations*.

List, J. A., Sadoff, S., and Wagner, M. (2011). So you want to run an experiment, now what? some simple rules of thumb for optimal experimental design. *Experimental Economics*, 14:439–457.

Machin, D., Cheung, Y. B., and Parmar, M. (2006). *Survival analysis: a practical approach*. John Wiley & Sons.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*.

Meta AI (2024). LLaMA 3.2 (1B, 3B). https://huggingface.co/meta-llama/Llama-3.2-1B.

Mohri, C. and Hashimoto, T. (2024). Language Models with Conformal Factuality Guarantees. In *International Conference on Machine Learning*.

Nagpal, C., Yadlowsky, S., Rostamzadeh, N., and Heller, K. (2021). Deep Cox Mixtures for Survival Regression. In *Machine Learning for Healthcare Conference*, pages 674–708. PMLR.

Papadopoulos, H. (2008). Inductive conformal prediction: Theory and application to neural networks. In *Tools in artificial intelligence*. Citeseer.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. In *European Conference on Machine Learning*.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. (2022). Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Prinster, D., Stanton, S., Liu, A., and Saria, S. (2024). Conformal Validity Guarantees Exist for Any Data Distribution (and How to Find Them). In *International Conference on Machine Learning*.

Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. (2024). Conformal Language Modeling. In *International Conference on Learning Representations*.

Rao, A., Vashistha, S., Naik, A., Aditya, S., and Choudhury, M. (2024). Tricking llms into disobedience: Formalizing, analyzing, and detecting jailbreaks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16802–16830. First released on arXiv in 2023.

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154.

Sesia, M. and Svetnik, V. (2025). Conformal survival bands for risk screening under right-censoring. *arXiv preprint arXiv:2505.04568*.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*.

Stroebl, B., Kapoor, S., and Narayanan, A. (2024). Inference Scaling fLaws: The Limits of LLM Resampling with Imperfect Verifiers. *arXiv preprint arXiv:2411.17501*.

Su, J., Kempe, J., and Ullrich, K. (2024). Mission impossible: A statistical perspective on jailbreaking llms. *Advances in Neural Information Processing Systems*, 37:38267–38306.

Tibshirani, R. J., Foygel Barber, R., Candès, E., and Ramdas, A. (2019). Conformal Prediction Under Covariate Shift. In *Advances in Neural Information Processing Systems*.

Vidgen, B., Scherrer, N., Kirk, H. R., Qian, R., Kannappan, A., Hale, S. A., and Röttger, P. (2023). SimpleSafetyTests: a Test Suite for Identifying Critical Safety Risks in Large Language Models. *arXiv preprint arXiv:2311.08370*.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*, volume 29. Springer.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. (2023). Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Wang, S., Jiang, Y., Tang, Y., Cheng, L., and Chen, H. (2025). Copu: Conformal prediction for uncertainty quantification in natural language generation. *arXiv preprint arXiv:2502.12601*.

Wang, Z., Duan, J., Cheng, L., Zhang, Y., Wang, Q., Shi, X., Xu, K., Shen, H., and Zhu, X. (2024). Conu: Conformal uncertainty in large language models with correctness coverage guarantees. *arXiv preprint arXiv:2407.00499*.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., and Poli, I. (2024). Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. *arXiv preprint arXiv:2412.13663*.

Westerlund, M. (2019). The emergence of deepfake technology: A review. *Technology innovation management review*, 9(11).

Xu, W., Wei, Z., Sun, X., Zhang, D., Yang, D., Zou, Q., and Zhang, X. (2025). Utilizing jailbreak probability to attack and safeguard multimodal llms. *arXiv preprint arXiv:2503.06989*.

Zrnic, T. and Candès, E. (2024). Active Statistical Inference. In *International Conference on Machine Learning*.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, all of our theory is given in Appendix D. Our algorithms are provided in Appendix B.]

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, the time complexity and data sizes are given in Appendix E.2]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, our code is anonymized.]

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes, all our theoretical statements contain the full set of assumptions, see Appendix D.]

(b) Complete proofs of all theoretical results. [Yes, all proofs are given in Appendix D.]

(c) Clear explanations of any assumptions. [Yes, we explain our assumptions throughout the paper.]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes, the full experimental details are provided in Appendix E.]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, all training details are provided in Appendix E.]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes, measures and statistics are provided in Section 6 and in Appendix E.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, see AppendixE.4.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes, we cited existing assets throughout the paper.]

(b) The license information of the assets, if applicable. [Yes, all assets used are publicly available.]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes, our code is provided in the supplemental material.]

(d) Information about consent from data providers/curators. [Not Applicable, all assets used are publicly available.]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes, see Section 7.]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable, we did not use crowdsourcing.]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable, we did not use crowdsourcing.]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable, we did not use crowdsourcing.]

A ADDITONAL RELATED WORK

The methods introduced in our paper draw on ideas from survival analysis, conformal prediction, and black-box uncertainty quantification. The `Optimized` method (Section 5.2) in particular is also related to: conformal prediction foundations, uncertainty quantification for LLMs, active learning and statistical inference, and optimal experimental design.

Conformal Prediction. The theoretical foundations of conformal prediction were established by Kolmogorov (1983, 1968); Papadopoulos et al. (2002); Vovk et al. (2005). This framework has since been extended to accommodate covariate shift by applying weights to the distribution of conformity scores (Tibshirani et al., 2019). Similar weighting strategies have also been employed to integrate active learning into conformal prediction (Prinster et al., 2024). Building further on these weighted approaches, conformal prediction methods have recently been adapted for survival analysis tasks (Candès et al., 2023; Gui et al., 2024a; Davidov et al., 2025).

Uncertainty Quantification for LLMs. Several recent works adapt conformal ideas to off-the-shelf LLMs. For instance, Wang et al. (2024) adapts conformal prediction to open-ended generation by integrating a self–consistency–based uncertainty measure, achieving strict correctness coverage across seven popular LLMs and diverse domains. The work of Wang et al. (2025) further extends this by explicitly inserting the ground truth into candidate outputs and using logit-based nonconformity scores to guarantee coverage over a wider range of error rates, though its reliance on logits raises calibration concerns when only API-level access is available. Additionally, *Conformal Language Modeling* has been developed by Quach et al. (2024) to calibrate both stopping and rejection rules to guarantee coverage in open-domain QA and summarization. Lin et al. (2024) distinguish uncertainty from confidence using semantic dispersion metrics, enabling selective generation of low-confidence outputs. To refine validity under practical constraints, Cherian et al. (2024) extended conditional conformal methods to incorporate utility-based guarantees and differentiable filtering. Mohri and Hashimoto (2024) introduced "conformal factuality," constructing entailment-based uncertainty sets that deliver high-probability correctness. The approach proposed in (Gui et al., 2024b) provides a guarantee for aligned responses in QA and radiology applications.

Active Learning and Statistical Inference. Active learning methods aim to select the most informative samples for labeling, improving model accuracy under tight budgets (Settles, 2010). This was formalized in (Zrnic and Candès, 2024) for statistical inference tasks, where one prioritizes labeling points with high model uncertainty while maintaining valid confidence intervals. Our `Optimized` method relates to this idea as it derives sample probabilities that directly minimize the variance of the augmented inverse-propensity-weighting estimator.

Optimal Experimental Design. Optimal experimental design allocates observations to minimize estimator variance under a given model. In (Hahn et al., 2011), the authors develop a two-stage adaptive design that chooses propensity scores to minimize the asymptotic variance bound in treatment-effect estimation, and the approach presented in (List et al., 2011) presents simple rules of thumb for efficient designs under resource constraints. Analogously, our `Optimized` method treats the sampling rule $\pi(x)$ as a design variable and solves for the allocation that minimizes the miscoverage estimation variance.

Classical Survival Analysis. The survival analysis problem was first formalized in biostatistics and reliability engineering. Kaplan and Meier (1958) introduce a nonparametric estimator of the survival function for right-censored data, laying the groundwork for all subsequent methods. The proportional hazards model was later developed, formalizing a semiparametric regression framework that relates covariates to an event's hazard rate without specifying its baseline form. As an alternative formulation, accelerated failure time (AFT) (Lin et al., 1998) models assume a parametric form for the log-survival time, allowing direct modeling of time-to-event via common distributions such as Weibull or log-normal. More recently, machine-learning approaches have been developed for survival prediction. Random survival forests leverage ensemble tree methods to nonparametrically estimate cumulative hazard functions (Ishwaran et al., 2008). DeepSurv employs a neural-network approximation of the Cox partial likelihood to capture complex covariate interactions (Katzman et al., 2018), and Deep Cox Mixtures generalizes this by modeling mixtures of proportional hazards to flexibly adapt to heterogeneous subpopulations (Nagpal et al., 2021).

B CALIBRATION ALGORITHMS

B.1 Calibration With a Known Censoring Mechanism

We present the general calibration with known censoring mechanism in Algorithm 1.

Algorithm 1 Calibration With a Known Censoring Mechanism

Require: Calibration data $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, censoring weights $\{w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\}_{i\in\mathcal{I}_2,\tau\in\mathcal{T}}$, quantile estimates $\{\hat{f}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, target miscoverage rate α, prior quantile τ_{prior}.

for $\tau \in \mathcal{T} \cap [0, \tau_{\text{prior}}]$ **do**
  $\hat{\alpha}(\tau) \leftarrow \frac{1}{|\mathcal{I}_2|} \sum_{i\in\mathcal{I}_2} w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\, \mathbb{I}\{\tilde{T}_i < \hat{f}_\tau(X_i) \leq C_i\}$ // miscoverage est.
end for
$\hat{\tau} \leftarrow \sup\Big\{\tau \in \mathcal{T} \cap [0, \tau_{\text{prior}}] : \sup_{\substack{\tau'\in\mathcal{T} \\ \tau'\leq\tau}} \hat{\alpha}(\tau') \leq \alpha\Big\}$ // calibrated quantile level
return Lower predictive bound (LPB) for a test point $X_{\text{test}} = x$, given by $\hat{L}(x) = \hat{f}_{\hat{\tau}}(x)$

B.2 Calibration With a Known Censoring Mechanism: `Naive` Mode

We describe the `Naive` calibration approach in Algorithm 4. First, we provide two auxiliary algorithms to sample outcomes from a generative model.

Algorithm 2 Generate and Audit Generative Model Responses

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, Generative model $\mathcal{G}(\cdot)$, audit function $\texttt{Audit}(\cdot)$, Censoring times $\{C_i\}_{i\in\mathcal{I}_2}$.
for each $i \in \mathcal{I}$ **do**
  Initialize $j \leftarrow 0$
  repeat
    $j \leftarrow j + 1$
    Generate and evaluate the output's safety: $Y_i^j \leftarrow \texttt{Audit}(X_i, \mathcal{G}(X_i))$
  until $Y_i^j = 1$ **or** $C_i = j$
  Set $\tilde{T}_i \leftarrow j$
end forreturn $\{\tilde{T}_i\}_{i\in\mathcal{I}_2}$

Algorithm 3 Generate Generative Model Outcomes

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, generative model $\mathcal{G}(\cdot)$, max per-prompt budget $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}}$, per-prompt evaluation probability $\{\pi_i\}_{i\in\mathcal{I}}$, audit function $\texttt{Audit}(\cdot)$.
for each $i \in \mathcal{I}$ **do**
  Draw $V_i \sim \text{Bernoulli}(\pi_i)$
  Set $C_i \leftarrow V_i \cdot \hat{f}_{\tau_{\text{prior}}}(X_i)$
end for
Obtain $\{\tilde{T}_i\}_{\{i\in\mathcal{I}_2\}}$ from Algorithm 2 applied with $\{X_i\}_{i\in\mathcal{I}_2}$, $\mathcal{G}$, $\{C_i\}_{i\in\mathcal{I}_2}$. **return** $\{(\tilde{T}_i, C_i)\}_{i\in\mathcal{I}_2}$.

B.3 Calibration With a Known Censoring Mechanism: `Basic` Mode

The calibration procedure with `Basic` budget allocation is outlined in Algorithm 5.

B.4 Calibration With a Known Censoring Mechanism: `Trimmed` Mode

We describe the calibration procedure with trimmed budget allocation in Algorithm 6.

Algorithm 4 Calibration With a Known Censoring Mechanism: `Naive` mode

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, generative model $\mathcal{G}(\cdot)$, audit function $\texttt{Audit}(\cdot)$, pre-trained quantile regression model $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, target miscoverage rate α, prior quantile τ_{prior}, total budget B.

Sample $C_i \sim \text{Geom}(n/B)$ for all $i \in \mathcal{I}_2$

Obtain $\{\tilde{T}_i\}_{\{i\in\mathcal{I}_2\}}$ from Algorithm 2 applied with $\{X_i\}_{i\in\mathcal{I}_2}$, $\mathcal{G}$, $\{C_i\}_{i\in\mathcal{I}_2}$.

$w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i) \leftarrow \frac{1}{\mathbb{P}[\hat{q}_\tau(X_i)\leq C_i|X_i]},\ i \in \mathcal{I}_2$. // compute the weights for the geometric C_i.

$\{(\tilde{T}_i, C_i)\}_{i\in\mathcal{I}_2} \leftarrow$ Algorithm 3 applied to $\{X_i\}_{i\in\mathcal{I}_2}$ with $\{\pi_i\}_{i\in\mathcal{I}_2}$ and $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$

$w(\{X_j\}_{j\in\mathcal{I}_2}, i) \leftarrow \frac{1}{\pi_i},\ i \in \mathcal{I}_2$

Obtain $\hat{L}(x)$ from Algorithm 1, applied with $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, $\{w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\}_{i\in\mathcal{I}_2}$, $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, α, and τ_{prior}.

return $\hat{L}(x)$, the lower predictive bound (LPB) for a test point $X_{\text{test}} = x$.

Algorithm 5 Calibration With a Known Censoring Mechanism: `Basic` mode

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, generative model $\mathcal{G}(\cdot)$, audit function $\texttt{Audit}(\cdot)$, pre-trained quantile regression model $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, target miscoverage rate α, prior quantile τ_{prior}, total budget B.

$\{\hat{f}_\tau(X_i)\}_{\tau\in\mathcal{T}, i\in\mathcal{I}_2} \leftarrow \{\hat{q}_\tau(X_i)\}_{\tau\in\mathcal{T}, i\in\mathcal{I}_2}$

$\{\pi_i\} \leftarrow \{\min(\frac{B}{|\mathcal{I}_2|\ \hat{f}_{\tau_{\text{prior}}}(X_i)}, 1)\}$ // per-prompt evaluation probability (Section 6.1)

$\{(\tilde{T}_i, C_i)\}_{i\in\mathcal{I}_2} \leftarrow$ Algorithm 3 applied to $\{X_i\}_{i\in\mathcal{I}_2}$ with $\{\pi_i\}_{i\in\mathcal{I}_2}$ and $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$

$w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i) \leftarrow \frac{1}{\pi_i},\ i \in \mathcal{I}_2$

Obtain $\hat{L}(x)$ from Algorithm 1, applied with $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, $\{w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\}_{i\in\mathcal{I}_2}$, $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, α, and τ_{prior}.

return $\hat{L}(x)$, the lower predictive bound (LPB) for a test point $X_{\text{test}} = x$.

Algorithm 6 Calibration With a Known Censoring Mechanism: `Trimmed` Mode

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, generative model $\mathcal{G}(\cdot)$, audit function $\texttt{Audit}(\cdot)$, pre-trained quantile regression model $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, target miscoverage rate α, prior quantile τ_{prior}, total budget B.

$\{\hat{f}_\tau(X_i)\}_{\tau\in\mathcal{T}, i\in\mathcal{I}_2} \leftarrow \{\hat{q}_\tau(X_i)\}_{\tau\in\mathcal{T}, i\in\mathcal{I}_2}$

$\{\pi_i\} \leftarrow \{\min(\frac{B}{|\mathcal{I}_2|\ \hat{f}_{\tau_{\text{prior}}}(X_i)}, 1)\}$ // per-prompt evaluation probability (Section 6.1)

$\{(\tilde{T}_i, C_i)\}_{i\in\mathcal{I}_2} \leftarrow$ Algorithm 3 applied to $\{X_i\}_{i\in\mathcal{I}_2}$ with $\{\pi_i\}_{i\in\mathcal{I}_2}$ and $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$

$w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i) \leftarrow \frac{1}{\pi_i},\ i \in \mathcal{I}_2$

Obtain $\hat{L}(x)$ from Algorithm 1, applied with $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, $\{w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\}_{i\in\mathcal{I}_2}$, $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, α, and τ_{prior}.

return $\hat{L}(x)$, the lower predictive bound (LPB) for a test point $X_{\text{test}} = x$.

B.5 Calibration With a Known Censoring Mechanism: `Optimized` Mode

In this section, we detail the optimized adaptive budgeting algorithm. First, observe that if the budget is sufficiently large such that the sum of $\hat{f}_{\tau_{\text{prior}}}(X_i)$ does not exceed the budget, i.e., $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) \leq B$, then we can set the censoring times to the target value $C_i = \hat{f}_{\tau_{\text{prior}}}(X_i)$ for all calibration points. In this case, the solution for (10) is straightforward, and the optimized probabilities are equal to one: $\pi_i^* = 1,\ \forall i \in \mathcal{I}_2$. Since this setup is trivial, we will consider the more challenging setting, in which the target values exceed the threshold $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) > B$.

Since each summand $1/\pi_i$ is strictly convex in π_i on $(0, 1]$ and the constraint is linear, (10) is a convex program. Furthermore, if the sum of $\hat{f}_{\tau_{\text{prior}}}(X_i)$ exceeds the budget, the constraint turns into an equality constraint, as stated next.

Proposition B.1. *Assume $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) > B$. Then any minimizer π^* of (10) must satisfy the budget constraint with equality:*

$$\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^* \;=\; B.$$

We now turn to show how to compute the optimized probabilities π^*. Under Proposition B.1, the optimization constraint in (10) turns into an equality constraint, and the Lagrangian is therefore given by

$$\mathcal{L}(\pi, \lambda) = \frac{1}{|\mathcal{I}_2|} \sum_{i\in\mathcal{I}_2} \frac{1}{\pi_i} + \lambda\Big(\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i - B\Big),$$

where $\lambda > 0$ is the Lagrange multiplier. On the open domain $(0, 1]^n$, the objective is strictly convex and differentiable, so the single stationary point satisfying the first-order conditions is in fact the unique global minimizer. Taking partial derivatives,

$$\frac{\partial\mathcal{L}}{\partial\pi_i} = -\frac{1}{|\mathcal{I}_2|\,\pi_i^2} + \lambda\,\hat{f}_{\tau_{\text{prior}}}(X_i) \;=\; 0 \quad\Longrightarrow\quad \pi_i = \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda\,\hat{f}_{\tau_{\text{prior}}}(X_i)}}.$$

Applying the box constraint $\pi_i \leq 1$ gives

$$\pi_i^*(\lambda) = \min\Big\{1,\ \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda\,\hat{f}_{\tau_{\text{prior}}}(X_i)}}\Big\}.$$

Now, define the budget-usage function

$$U(\lambda) \;=\; \sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^*(\lambda).$$

The above equation has a unique solution $\lambda^* > 0$ which satisfies $U(\lambda^*) = B$. This $\lambda^* > 0$ can be recovered by bisection, across the following interval:

$$\left[\lambda_{\text{low}} = \frac{1}{|\mathcal{I}_2|\max_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)},\ \lambda_{\text{high}} = \frac{|\mathcal{I}_2|\max_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)^2}{B^2 \min_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)}\right]. \tag{11}$$

This procedure is summarized in Algorithm 8 and it produces a valid solution, as stated in the following proposition.

Proposition B.2. *The output π^* of Algorithm 8 is the unique solution of (10) which satisfies $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^* \leq B$.*

Algorithm 7 Calibration with a known censoring mechanism: optimized budget allocation

Require: Calibration data $\{X_i\}_{i\in\mathcal{I}_2}$, generative model $\mathcal{G}(\cdot)$, audit function $\texttt{Audit}(\cdot)$, pre-trained quantile regression model $\{\hat{q}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, target miscoverage rate α, prior quantile τ_{prior}, quantile trimming threshold M, total budget B.

$\{\hat{f}_\tau(X_i)\}_{\tau\in\mathcal{T},i\in\mathcal{I}_2} \leftarrow \{\min(\hat{q}_\tau(X_i), M)\}_{\tau\in\mathcal{T},i\in\mathcal{I}_2}$ // trim the quantile est. (Section 6.1)
$\{\pi_i\} \leftarrow$ Algorithm 8 applied with $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$ // optimized per-prompt evaluation probability (Section 5.2)
$\{(\tilde{T}_i, C_i)\}_{i\in\mathcal{I}_2} \leftarrow$ Algorithm 3 applied to $\{X_i\}_{i\in\mathcal{I}_2}$ with $\{\pi_i\}_{i\in\mathcal{I}_2}$ and $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$
$w(\{X_j\}_{j\in\mathcal{I}_2}, i) \leftarrow \frac{1}{\pi_i},\ i \in \mathcal{I}_2$
Obtain $\hat{L}(x)$ from Algorithm 1, applied with $\{X_i, C_i, \tilde{T}_i\}_{i\in\mathcal{I}_2}$, $\{w_i,\}_{i\in\mathcal{I}_2}$, $\{\hat{f}_\tau(\cdot)\}_{\tau\in\mathcal{T}}$, α, and τ_{prior}.
return $\hat{L}(x)$, the lower predictive bound (LPB) for a test point $X_{\text{test}} = x$.

Algorithm 8 Get Optimal Per-Prompt Evaluation Probabilities

Require: effective sample cost $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$, budget B, bijection tolerance ϵ, initial $\lambda_{low}, \lambda_{high}$

Ensure: Optimal π^* and multiplier λ^* satisfying $\pi_i^* = \min\{1, 1/\sqrt{\lambda^* \hat{f}_{\tau_{\text{prior}}}(X_i)}\}$ and $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^* = B$.

if $B > \sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)$ **then**
  $\pi^* \leftarrow [1, \ldots, 1]$, $\lambda^* \leftarrow \texttt{Null}$
else
  while $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^*(\lambda_{high}) > B$ **do**
    $\lambda_{\text{high}} \leftarrow 2\,\lambda_{\text{high}}$
  end while
  $\lambda^* \leftarrow$ bisection on $[\lambda_{\text{low}}, \lambda_{\text{high}}]$ to solve $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^*(\lambda) = B$ within ϵ
  $\pi_i^* \leftarrow \min\{1, 1/\sqrt{\lambda^* \hat{f}_{\tau_{\text{prior}}}(X_i)}\}$ for all $i \in \mathcal{I}_2$
end ifreturn (π^*, λ^*)

C A DEEPER LOOK INTO $\hat{\alpha}(\tau)$

In Section 4, we introduced the miscoverage estimator $\hat{\alpha}(\tau)$. In this section, we explore its statistical properties in detail. We begin by deriving its expectation and variance:

Proposition C.1 (Unbiasedness and conditional variance of the weighted miscoverage estimator)**.** *Under the conditional independence assumption 2.1 in Section 2, the estimator $\hat{\alpha}(\tau)$ defined in equation* (6) *satisfies*

$$\mathbb{E}\big[\hat{\alpha}(\tau)\big] = \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} \mathbb{P}\big[T_i < \hat{q}_\tau(X_i)\big].$$

If we further assume that for all $i \neq i' \in \mathcal{I}_2$, $(C_i, T_i) \perp\!\!\!\perp (C_{i'}, T_{i'}) \mid \{X_j\}_{j\in\mathcal{I}_2}$, we have

$$\mathrm{Var}\big[\hat{\alpha}(\tau) \mid \{X_j\}_{j\in\mathcal{I}_2}\big] = \frac{1}{|\mathcal{I}_2|^2}\sum_{i\in\mathcal{I}_2}\Big\{w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \mid \{X_j\}_{j\in\mathcal{I}_2}\big] - \big[\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \mid \{X_j\}_{j\in\mathcal{I}_2}\big]\big]^2\Big\}.$$

See Appendix D.1 for the proof. We remark that the conditional independence and pair-wise independence assumptions in Proposition C.1 hold for all variants of our method from Sections 5. This proposition shows that $\hat{\alpha}(\tau)$ is an unbiased miscoverage estimator, resulting in a useful approximation for the miscoverage estimation variance. Under a simplified assumption that, for a fixed τ, each calibration point $i \in \mathcal{I}_2$ is miscovered at the same constant rate conditional on $\{X_j\}_{j\in\mathcal{I}_2}$, we have the following result.

Proposition C.2 (Variance linearly increasing in mean weight under constant miscoverage)**.** *Under Assumption 2.1, and additionally assuming that for all $i \neq i'$, $(C_i, T_i) \perp\!\!\!\perp (C_{i'}, T_{i'}) \mid \{X_j\}_{j\in\mathcal{I}_2}$, and that for any fixed τ,*

each calibration point $i \in \mathcal{I}_2$ is miscovered at the same constant rate conditional on $\{X_j\}_{j\in\mathcal{I}_2}$,

$$\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] = \mathrm{Const}_\tau.$$

Then, the variance of $\hat{\alpha}(\tau)$ from (6) is

$$\mathrm{Var}\big[\hat{\alpha}(\tau) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] = \frac{\mathrm{Const}_\tau}{|\mathcal{I}_2|}\overline{w}_\tau - \frac{\mathrm{Const}_\tau^2}{|\mathcal{I}_2|},$$

where $\overline{w}_\tau = |\mathcal{I}_2|^{-1}\sum_{i\in\mathcal{I}_2} w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)$ is the mean weight.

See proof in Appendix D.2.

D PROOFS

D.1 Proof of Proposition C.1

Proof. Recall that

$$\hat{\alpha}(\tau) = \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} w(X_i)\,\mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\}\,\mathbb{I}\{T_i < \hat{q}_\tau(X_i)\}.$$

Set

$$V_i = w(X_i)\,\mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\}\,\mathbb{I}\{T_i < \hat{q}_\tau(X_i)\}.$$

By Assumption 2.1 each of C_i and T_i is independent of the other given X_i. Hence for each i,

$$\begin{aligned}
\mathbb{E}\big[V_i \mid \{X_j\}_{j\in\mathcal{I}_2}\big] &= \mathbb{E}\big[w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\}\,\mathbb{I}\{T_i < \hat{q}_\tau(X_i)\} \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\
&= \mathbb{E}\big[w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\} \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]\,\mathbb{E}\big[\mathbb{I}\{T_i < \hat{q}_\tau(X_i)\} \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\
&= w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{P}\big[\hat{q}_\tau(X_i) \le C_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]\,\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\
&= \mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big],
\end{aligned}$$

where the first is by the tower property, and in the last transition we used $w(\{X_j\}_{j\in\mathcal{I}_2}, i) = \mathbb{P}\big[\hat{q}_\tau(X_i) \le C_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]^{-1}$, from (5).

Consequently,

$$\begin{aligned}
\mathbb{E}[\hat{\alpha}(\tau)] &= \mathbb{E}\Big[\tfrac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} V_i\Big] \\
&= \mathbb{E}\Big[\tfrac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} \mathbb{E}\big[V_i \mid \{X_j\}_{j\in\mathcal{I}_2}\big]\Big] \\
&= \mathbb{E}\Big[\tfrac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} \mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \mid X_i\big]\Big] = \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} \mathbb{P}\big[T_i < \hat{q}_\tau(X_i)\big].
\end{aligned}$$

For the conditional variance, we similarly derive

$$\begin{aligned}
\mathbb{E}\big[V_i^2 \mid \{X_j\}_{j\in\mathcal{I}_2}\big] &= \mathbb{E}\big[w_\tau^2(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{I}\{\hat{q}_\tau(X_i) \le C_i\}\,\mathbb{I}\{T_i < \hat{q}_\tau(X_i)\} \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\
&= w_\tau^2(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{P}\big[\hat{q}_\tau(X_i) \le C_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]\,\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\
&= w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big].
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathrm{Var}\big[V_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] =& \mathbb{E}\big[V_i^2 \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] - \big(\mathbb{E}[V_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}]\big)^2 \\
=& w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}\big] - \Big[\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}\big]\Big]^2.
\end{aligned}$$

Conditioned on $\{X_j\}_{j\in\mathcal{I}_2}$, the pairs (C_i, T_i) are independent across i, and so for $i \neq j$,

$$\mathrm{Cov}(V_i, V_j \mid \{X_j\}) = 0.$$

Consequently,

$$\begin{aligned}\mathrm{Var}\big[\hat{\alpha}(\tau) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] =& \mathrm{Var}\Big[\tfrac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} V_i \;\Big|\; \{X_j\}_{j\in\mathcal{I}_2}\Big] \\ =& \tfrac{1}{|\mathcal{I}_2|^2}\sum_{i\in\mathcal{I}_2}\Big\{w(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] \\ &- \big[\mathbb{P}\big[T_i < \hat{q}_\tau(X_i) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]\big]^2\Big\}\end{aligned}$$

□

D.2 Proof of Proposition C.2

Proof of Proposition C.2. Under the proposition's assumptions, Proposition C.1 shows us that the variance of $\hat{\alpha}(\tau)$ simplifies to

$$\begin{aligned}\mathrm{Var}\big[\hat{\alpha}(\tau) \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big] &= \frac{1}{|\mathcal{I}_2|^2}\sum_{i\in\mathcal{I}_2}\Big\{w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\,\mathrm{Const}_\tau - \mathrm{Const}_\tau^2\Big\} \\ &= \frac{\mathrm{Const}_\tau}{|\mathcal{I}_2|}\overline{w}_\tau - \frac{\mathrm{Const}_\tau^2}{|\mathcal{I}_2|}\end{aligned}$$

where $\overline{w}_\tau = |\mathcal{I}_2|^{-1}\sum_{i\in\mathcal{I}_2} w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)$ is the mean weight. □

D.3 Proof of Proposition B.1

Proof of Proposition B.1. Since

$$\frac{\partial}{\partial\pi_i}\Big(\frac{1}{|\mathcal{I}_2|}\sum_{j\in\mathcal{I}_2}\tfrac{1}{\pi_j}\Big) = -\frac{1}{|\mathcal{I}_2|\,\pi_i^2} < 0,$$

increasing any single π_i (while keeping the others fixed) strictly decreases the objective. Assume for the sake of contradiction that there exist an optimal solution for (10), π^*, satisfying $\sum_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^* < B$. Then, there exist an index $j \in \mathcal{I}_2$ and $\delta > 0$ such that $\pi_j^* + \delta \leq 1$ and $\sum_i \hat{f}_{\tau_{\text{prior}}}(X_i)\,\tilde{\pi}_i \leq B$, where $\tilde{\pi}_i = \pi_i^*$ for $i \neq j$ and $\tilde{\pi}_j = \pi_j^* + \delta$. But then

$$\frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}\frac{1}{\tilde{\pi}_i} \;<\; \frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}\frac{1}{\pi_i^*},$$

contradicting optimality of π^*. Therefore, the budget must bind:

$$\sum_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i)\,\pi_i^* = B.$$

□

D.4 Proof of Proposition B.2

Proof of Proposition B.2. Notice that if $\sum_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i) \leq B$, then $\pi_i^* = 1$ for all $i \in \mathcal{I}_2$ is the optimal solution which satisfies the constraint. We now turn to consider the case where $\sum_{i\in\mathcal{I}_2}\hat{f}_{\tau_{\text{prior}}}(X_i) > B$. We first show that there exists a unique solution λ^*. Observe that $U : (0,\infty) \to (0, \sum_i \hat{f}_{\tau_{\text{prior}}}(X_i)]$ is continuous and strictly decreasing, with $\lim_{\lambda\to 0^+} U(\lambda) = \sum_i \hat{f}_{\tau_{\text{prior}}}(X_i) > B$ and $\lim_{\lambda\to\infty} U(\lambda) = 0$. By the intermediate-value theorem there is a unique $\lambda^* > 0$ such that $U(\lambda^*) = B$. Next, the output of Algorithm 8 is indeed the solution λ^* since U is a monotone function, and thus the bisection approach is valid in this case. Therefore, $\pi^*(\lambda)$ is the correct optimal solution. □

D.5 Proof of Proposition 5.1

Proof of Proposition 5.1. First, notice that by assuming $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) \leq B$, is follows that $\pi_i^* = 1$ for all $i \in \mathcal{I}_2$ is the optimal solution which satisfies the constraint. We now turn to analyze the setup where $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) > B$. We prove by induction over the sum $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i)$, assuming that $\hat{f}_{\tau_{\text{prior}}}(X_i) \leq M \ \forall i \in \mathcal{I}_2$.

Base case. We begin with the maximal sum: $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) = |\mathcal{I}_2|M$. That is, $\hat{f}_{\tau_{\text{prior}}}(X_i) = M$ for all $i \in \mathcal{I}_2$. Then, by the optimization constraint, we get

$$M \sum_{i\in\mathcal{I}_2} \pi_i^* = B \qquad \Longrightarrow \qquad \sum_{i\in\mathcal{I}_2} \pi_i^* = \frac{B}{M},$$

and by symmetry the unique minimizer of $\sum_i 1/\pi_i^*$ with $\sum_i \pi_i^* = B/M$ is

$$\pi_i^* = \frac{B}{n\,M} \quad \forall i \in \mathcal{I}_2.$$

Inductive step. We now suppose that the claim holds for any weight vector with a sum $B - 1 < S \leq M|\mathcal{I}_2|$, and show that it holds for any vector with a sum $S - 1$. Given a weight vector $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$ with a sum $\sum_{i\in\mathcal{I}_2} \hat{f}_{\tau_{\text{prior}}}(X_i) = S - 1$, we compose a new vector with a sum S, as follows. In $\{\hat{f}_{\tau_{\text{prior}}}(X_i)\}_{i\in\mathcal{I}_2}$, there exists an index i_0 such that $\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) \leq M - 1$. The new vector with a sum S is defined by:

$$\hat{f}'_{\tau_{\text{prior}}}(X_i) = \begin{cases} \hat{f}_{\tau_{\text{prior}}}(X_i), & i \neq i_0, \\ \hat{f}_{\tau_{\text{prior}}}(X_i) + 1, & i = i_0, \end{cases}$$

Notice that $\max_i \hat{f}'_{\tau_{\text{prior}}}(X_i) \leq M$ and $\sum_{i\in\mathcal{I}_2} \hat{f}'_{\tau_{\text{prior}}}(X_i) = S$. Let λ and λ' be the unique solutions of

$$U(\lambda) = B \quad \text{and} \quad U'(\lambda') = B,$$

where

$$U'(\lambda) = \sum_{i\neq i_0} \hat{f}_{\tau_{\text{prior}}}(X_i) \min\Big\{1, \tfrac{1}{\sqrt{n\lambda \hat{f}_{\tau_{\text{prior}}}(X_i)}}\Big\} + (\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1) \min\Big\{1, \tfrac{1}{\sqrt{n\lambda(\hat{f}_{\tau_{\text{prior}}}(X_{i_0})+1)}}\Big\}.$$

Since for every $\lambda > 0$ we have

$$\frac{\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1}{\sqrt{|\mathcal{I}_2|\lambda(\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1)}} = \frac{\sqrt{\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1}}{\sqrt{|\mathcal{I}_2|\lambda}} > \frac{\sqrt{\hat{f}_{\tau_{\text{prior}}}(X_{i_0})}}{\sqrt{|\mathcal{I}_2|\lambda}} = \frac{\hat{f}_{\tau_{\text{prior}}}(X_{i_0})}{\sqrt{|\mathcal{I}_2|\lambda \hat{f}_{\tau_{\text{prior}}}(X_{i_0})}},$$

and of course $\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1 > \hat{f}_{\tau_{\text{prior}}}(X_{i_0})$, we get the inequality

$$(\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1) \min\Big\{1, \tfrac{1}{\sqrt{|\mathcal{I}_2|\lambda(\hat{f}_{\tau_{\text{prior}}}(X_{i_0})+1)}}\Big\} > \hat{f}_{\tau_{\text{prior}}}(X_{i_0}) \min\Big\{1, \tfrac{1}{\sqrt{|\mathcal{I}_2|\lambda \hat{f}_{\tau_{\text{prior}}}(X_{i_0})}}\Big\},$$

and thus,

$$\begin{aligned} U'(\lambda) &= \sum_{i\neq i_0} \hat{f}_{\tau_{\text{prior}}}(X_i)\, \pi_i^*(\lambda) + (\hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1) \min\Big\{1, \tfrac{1}{\sqrt{|\mathcal{I}_2|\lambda(\hat{f}_{\tau_{\text{prior}}}(X_{i_0})+1)}}\Big\} \\ &> \sum_{i\neq i_0} \hat{f}_{\tau_{\text{prior}}}(X_i)\, \pi_i^*(\lambda) + \hat{f}_{\tau_{\text{prior}}}(X_{i_0}) \min\Big\{1, \tfrac{1}{\sqrt{|\mathcal{I}_2|\lambda \hat{f}_{\tau_{\text{prior}}}(X_{i_0})}}\Big\} \\ &= U(\lambda) = B. \end{aligned}$$

Since U' is continuous and decreasing on $(0, \infty)$ with $\lim_{\mu\to 0^+} U'(\mu) = \sum_i \hat{f}_{\tau_{\text{prior}}}(X_i) > B$ and $\lim_{\mu\to\infty} U'(\mu) = 0$, the intermediate-value theorem guarantees a unique λ' with $U'(\lambda') = B$. Moreover, since U' is decreasing, and

$U'(\lambda) \geq B = U'(\lambda')$ we get $\lambda' \geq \lambda$. Define π_i the optimal solution of (10) with $\hat{f}_{\tau_{\text{prior}}}(X_i)$ and π'_i the optimal solution with $\hat{f}'_{\tau_{\text{prior}}}(X_i)$.

- If $i \neq i_0$, since $\hat{f}'_{\tau_{\text{prior}}}(X_i) = \hat{f}_{\tau_{\text{prior}}}(X_i)$,

$$\pi'_i = \min\Big\{1, \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda'\,\hat{f}'_{\tau_{\text{prior}}}(X_i)}}\Big\} \;\leq\; \min\Big\{1, \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda\,\hat{f}_{\tau_{\text{prior}}}(X_i)}}\Big\} = \pi_i.$$

- If $i = i_0$, then $\hat{f}'_{\tau_{\text{prior}}}(X_{i_0}) = \hat{f}_{\tau_{\text{prior}}}(X_{i_0}) + 1 > \hat{f}_{\tau_{\text{prior}}}(X_{i_0})$, so

$$\pi'_{i_0} = \min\Big\{1, \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda'\,(\hat{f}_{\tau_{\text{prior}}}(X_{i_0})+1)}}\Big\} \;\leq\; \min\Big\{1, \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda'\,\hat{f}_{\tau_{\text{prior}}}(X_{i_0})}}\Big\} \;\leq\; \min\Big\{1, \frac{1}{\sqrt{|\mathcal{I}_2|\,\lambda\,\hat{f}_{\tau_{\text{prior}}}(X_{i_0})}}\Big\} = \pi_{i_0}.$$

By the inductive assumption, $\pi'_i \geq B/(|\mathcal{I}_2|M)$ for all $i \in \mathcal{I}_2$, hence $\pi_i \geq B/(|\mathcal{I}_2|M)$ for all $i \in \mathcal{I}_2$ as well. Lastly, by the definition of the weights, we have:

$$w(\{X_j\}_{j\in\mathcal{I}_2}, i) = (\pi^*_i)^{-1} \leq \max\left(\frac{|\mathcal{I}_2|M}{B}, 1\right). \tag{12}$$

This completes the proof. □

D.6 Proofs of the Coverage Rate Guarantees

In this section, we present and prove the formal version Theorem 4.1 which builds on the proof of (Gui et al., 2024a, Theorem 3).

Theorem D.1 (General validity, formal)**.** *Fix a tolerance level $\delta \in (0,1)$ and a miscoverage level $\tau \in (0,1)$. Suppose that $\{(X_i, T_i)\}_{i=1,\ldots,n}$ and $(X_{\text{test}}, T_{\text{test}})$ are drawn i.i.d., and that the censoring times satisfy the conditional independence assumption (Assumption 2.1) and $(C_i, T_i) \perp\!\!\!\perp (C_j, T_j) | \{X_k\}_{k\in\mathcal{I}_2}$ for all $i \neq j \in \mathcal{I}_2$. We suppose that $\hat{q}_\tau(x)$ is non-decreasing and continuous in τ. Further, assume that the weights are computed using the true probabilities:*

$$w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i) = 1/\mathbb{P}\big[\hat{q}_\tau(X_i) \leq C_i \;\big|\; \{X_j\}_{j\in\mathcal{I}_2}\big]. \tag{13}$$

Above, $\hat{q}_\tau(X_i)$ can be either the estimated quantile of $T_i \mid X = X_i$ presented in Section 3, or its trimmed version $\hat{f}_\tau(X_i)$ from Section 6.1. We assume that there exists a constant $\gamma_\tau > 0$ such that the weights satisfy $w_\tau(x) \leq \gamma_\tau$ for P_X-almost all x. Consider the following estimated miscoverage rate:

$$\hat{\alpha}(\tau) = \frac{1}{|\mathcal{I}_2|} \sum_{i\in\mathcal{I}_2} w_\tau(\{X_j\}_{j\in\mathcal{I}_2}, i)\, \mathbb{I}\big\{\tilde{T}_i < \hat{q}_\tau(X_i) \leq C_i\big\} \tag{14}$$

and denote the calibrated quantile level by

$$\hat{\tau} = \sup\Big\{\tau \in \mathcal{T} : \sup_{\substack{\tau'\in\mathcal{T}\\ \tau'\leq\tau}} \hat{\alpha}(\tau') \leq \alpha\Big\}. \tag{15}$$

Above, the search space $\mathcal{T}$ is formulated as in Section 4, or using $\mathcal{T} \cap [0, \tau_{\text{prior}}]$, as in the proposed ***Adaptive*** *method from Section 6.1. We remark that for $\hat{\tau}$ to be well defined, we assume there exists $\tau' \in \mathcal{T}$ such that $\hat{\alpha}(\tau') \leq \alpha$. This assumption can be trivially satisfied by setting $\hat{q}_0(X_i) = 0$. Then, with probability at least $1-\delta$ over the draws of $\mathcal{D}$, the LPB $\hat{L}(x) = \hat{q}_{\hat{\tau}}(x)$ satisfies*

$$\mathbb{P}\left[T_{\text{test}} \geq \hat{L}(X_{\text{test}}) | \mathcal{D}\right] \geq 1 - \alpha - \sup_{\tau\in[0,1]} \left\{\sqrt{\frac{2\gamma_\tau^2 + 5}{|\mathcal{I}_2|} \cdot \log\left(\frac{1}{\delta}\right)}\right\}. \tag{16}$$

Proof. For ease of notation, we define the coverage gap by:

$$\Delta := \sup_{\tau\in[0,1]} \left\{\sqrt{\frac{2\gamma_\lambda^2 + 5}{|\mathcal{I}_2|} \cdot \log\left(\frac{1}{\delta}\right)}\right\}. \tag{17}$$

We define the oracle miscoverage level by:

$$\tau(\alpha+\Delta) = \sup\left\{\lambda \in [0,1] : \mathbb{P}\left(T < \hat{q}_\lambda(X) \mid \mathcal{I}_1\right) \leq \alpha + \Delta\right\}. \tag{18}$$

Observe that by assuming $1-\delta \leq \mathbb{P}(\hat{\tau} \leq \tau(\alpha+\Delta) \mid \mathcal{I}_1)$, we get that the event $\{\hat{\tau} \leq \tau(\alpha+\Delta)\}$ holds with probability at least $1-\delta$. Under the monotonicity of $\hat{q}_\tau$, and following the the left-continuity of $\mathbb{P}(T \geq \hat{q}_\tau(X) \mid \mathcal{D})$ in τ we obtain that with probability at least $1-\delta$:

$$\begin{aligned} &\mathbb{P}(T \geq \hat{q}_{\hat{\tau}}(X) \mid \mathcal{D}) \\ \geq &\mathbb{P}(T \geq \hat{q}_{\tau(\alpha+\Delta)}(X) \mid \mathcal{D}) \\ \geq &1-\alpha-\Delta. \end{aligned} \tag{19}$$

Now, we turn to show that $1-\delta \leq \mathbb{P}(\hat{\tau} \leq \tau(\alpha+\Delta) \mid \mathcal{I}_1)$. We begin by fixing $\varepsilon > 0$, and denoting $\lambda := \tau(\alpha+\Delta)+\varepsilon$. By the definition of $\hat{\alpha}(\tau)$, we get:

$$\begin{aligned} &\mathbb{P}(\hat{\alpha}(\tau(\alpha+\Delta)+\varepsilon) \leq \alpha \mid \mathcal{I}_1) \\ =&\mathbb{P}(\hat{\alpha}(\lambda) \leq \alpha \mid \mathcal{I}_1) \\ =&\mathbb{P}\left(\frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)\, \mathbb{I}\{\tilde{T}_i < \hat{q}_\tau(X_i) \leq C_i\} \leq \alpha \middle| \mathcal{I}_1\right) \\ =&\mathbb{P}\left(\frac{1}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)\, \mathbb{I}\{\hat{q}_\lambda(X_i) \leq C_i\}\, \mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\} - \alpha \leq 0 \middle| \mathcal{I}_1\right) \end{aligned} \tag{20}$$

Next, we apply Markov's inequality for any $t > 0$, and obtain:

$$(20) \leq \mathbb{E}\left(\exp\left(\alpha - \frac{t}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)\, \mathbb{I}\{\hat{q}_\lambda(X_i) \leq C_i\}\, \mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}\right) \middle| \mathcal{I}_1\right) \tag{21}$$

By conditioning on $\{(X_i, T_i)\}_{i\in\mathcal{I}_2}$ and following the $\frac{1}{4}$ sub-gaussianity of $w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)^{-1} - \mathbb{I}\{\hat{q}_\lambda(X_i) \leq C_i\}$, the conditionally independent censoring assumption (Assumption 2.1) and the bounded weights, we have:

$$\begin{aligned} &\mathbb{E}\left(\exp\left(\frac{t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)\mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}\left(w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)^{-1} - \mathbb{I}\{\hat{q}_\lambda(X_i) \leq C_i\}\right)\right) \middle| \{(X_i, T_i)\}_{i\in\mathcal{I}_2}, \mathcal{I}_1\right) \\ \leq &\exp\left(\frac{t^2}{8|\mathcal{I}_2|^2}\cdot\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)^2\mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}^2\right) \\ \leq &\exp\left(\frac{t^2}{8|\mathcal{I}_2|^2}\cdot\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)^2\right) \\ \leq &\exp\left(\frac{\gamma_\lambda^2 t^2}{8|\mathcal{I}_2|}\right). \end{aligned} \tag{22}$$

Therefore,

$$\begin{aligned} (21) &\leq \exp\left(\frac{\gamma_\lambda^2 t^2}{8|\mathcal{I}_2|}\right)\cdot\mathbb{E}\left(\exp\left(\alpha - \frac{t}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2} w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)\, w_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)^{-1}\, \mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}\right) \middle| \mathcal{I}_1\right) \\ &= \exp\left(\frac{\gamma_\lambda^2 t^2}{8|\mathcal{I}_2|}\right)\cdot\mathbb{E}\left(\exp\left(\alpha - \frac{t}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}\mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}\right) \middle| \mathcal{I}_1\right) \end{aligned} \tag{23}$$

Further, defining $p_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i) := \mathbb{P}(T_i < \hat{q}_\lambda(X_i) \mid \{X_j\}_{j\in\mathcal{I}_2}, \mathcal{I}_1)$, we condition on $\{X_j\}_{j\in\mathcal{I}_2}$ and get, using the

$\frac{1}{4}$ subgaussianity of $p_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i) - \mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}$:

$$\begin{aligned}
&\mathbb{E}\left(\exp\left(p_\lambda(\{X_j\}_{j\in\mathcal{I}_2}, i)) - \frac{t}{|\mathcal{I}_2|}\sum_{i\in\mathcal{I}_2}(\mathbb{I}\{T_i < \hat{q}_\lambda(X_i)\}\right)\Bigg|\{X_j\}_{j\in\mathcal{I}_2}, \mathcal{I}_1\right) \\
&\leq \exp\left(\frac{t^2}{8|\mathcal{I}_2|^2}\cdot|\mathcal{I}_2|\right) \\
&= \exp\left(\frac{t^2}{8|\mathcal{I}_2|}\right)
\end{aligned} \tag{24}$$

We plug this into (23) to obtain:

$$(23) \leq \exp\left(\frac{(\gamma_\lambda^2+1)t^2}{8|\mathcal{I}_2|}\right)\mathbb{E}\left(\exp\left(\frac{t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\alpha - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right) \tag{25}$$

We now apply the Cauchy-Schwarz inequality:

$$\begin{aligned}
&\mathbb{E}\left(\exp\left(\frac{t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\alpha - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right) \\
\leq&\mathbb{E}\left(\exp\left(\frac{2t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\alpha - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right)^{1/2}
\end{aligned} \tag{26}$$

By the definition of $\tau(\alpha+\Delta)$ we have:

$$\mathbb{P}(T < \hat{q}_\lambda(X) \mid \mathcal{I}_1) \geq \alpha + \Delta. \tag{27}$$

By plugging it into the above, we get:

$$\begin{aligned}
(26) =& \mathbb{E}\left(\exp\left(\frac{2t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\alpha - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right) \\
\leq&\mathbb{E}\left(\exp\left(\frac{2t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\mathbb{P}(T < \hat{q}_\lambda(X) \mid \mathcal{I}_1) - \Delta - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right) \\
=& \exp(-2t\Delta)\mathbb{E}\left(\exp\left(\frac{2t}{|\mathcal{I}_2|}\cdot\sum_{i\in\mathcal{I}_2}(\mathbb{P}(T < \hat{q}_\lambda(X) \mid \mathcal{I}_1) - p_\lambda(X_i))\right)\Bigg|\mathcal{I}_1\right) \\
\leq& \exp\left(\frac{t^2}{2|\mathcal{I}_2|} - 2t\Delta\right).
\end{aligned} \tag{28}$$

Above, we used the $\frac{1}{4}$-sub-gaussianity of $\mathbb{P}(T < \hat{q}_\lambda(X) \mid \mathcal{I}_1) - p_\lambda(X_i)$. By combining it all, we obtain:

$$\begin{aligned}
(20) \leq& \exp\left(\frac{t^2}{2|\mathcal{I}_2|} - 2t\Delta + \frac{(\gamma_\lambda^2+1)t^2}{8|\mathcal{I}_2|} + \frac{\gamma_\lambda^2 t^2}{8|\mathcal{I}_2|}\right) \\
=& \exp\left(\frac{(2\gamma_\lambda^2+5)t^2}{8|\mathcal{I}_2|} - 2t\Delta\right)
\end{aligned} \tag{29}$$

We define

$$t = \frac{(8+2\sqrt{14})|\mathcal{I}_2|\Delta}{2\gamma_\lambda^2+5} \tag{30}$$

and plug it into (25):

$$
\begin{aligned}
(25) \leq & \exp\left(\frac{(2\gamma_\lambda^2+5)(\frac{(8+2\sqrt{14})|\mathcal{I}_2|\Delta}{2\gamma_\lambda^2+5})^2}{8|\mathcal{I}_2|} - \frac{(16+4\sqrt{14})|\mathcal{I}_2|\Delta}{2\gamma_\lambda^2+5}\Delta\right) \\
= & \exp\left(\frac{(15+4\sqrt{14})|\mathcal{I}_2|\Delta^2}{(2\gamma_\lambda^2+5)} - \frac{(16+4\sqrt{14})}{2\gamma_\lambda^2+5}|\mathcal{I}_2|\Delta^2\right) \\
= & \exp\left(-\frac{1|\mathcal{I}_2|\Delta^2}{2\gamma_\lambda^2+5}\right) \\
\leq & \exp\left(-\log\left(\frac{1}{\delta}\right)\right) \\
= & \delta
\end{aligned}
\tag{31}
$$

That is, we just showed that:

$$
1-\delta \leq \mathbb{P}(\hat{\alpha}(\tau(\alpha+\Delta)+\varepsilon) > \alpha \mid \mathcal{I}_1) \leq \mathbb{P}(\hat{\tau} < \tau(\alpha+\Delta)+\varepsilon \mid \mathcal{I}_1). \tag{32}
$$

The above equation holds for every $\varepsilon > 0$, and thus by taking $\varepsilon \to 0$, and following the continuity of the probability measure, we obtain that $1-\delta \leq \mathbb{P}(\hat{\tau} \leq \tau(\alpha+\Delta) \mid \mathcal{I}_1)$, as required. □

Since the proposed methods construct censoring times that satisfy the conditional independence assumption of Theorem D.1, their validity follow immediately from this theorem with the corresponding definitions of the censoring times and the weights.

E EXPERIMENTAL DETAILS

This section provides additional information about the experimental setup and methodology described in Section 6. Section E.1 details the synthetic experiments, including data generation and model architecture. Section E.2 presents experiments conducted on real-world data. Finally, Section E.3 outlines implementation details that are shared across both experimental settings.

E.1 Synthetic Experiments

Synthetic Covariate Generation. As described in Section 6.2 of the main text, we use synthetic data to evaluate the informativity and coverage of the LPBs constructed by our calibration procedures. Each synthetic data point consists of covariates $X_i \in \mathbb{R}^d$ with $d = 10$, and a prompt-specific unsafe-output probability p_i. The dataset is designed to simulate a setting in which most prompts have a high likelihood of producing unsafe outputs, while a smaller subset is relatively safe. Notably, our experiments show that fitting a model on this synthetic dataset results in a large prediction error, which turns the calibration step more challenging.

To this end, we sample the covariates from a Gaussian distribution:

$$
X_i \mid p_i \sim \mathcal{N}\left(\mu(p_i), \sigma^2 I_d\right), \quad \text{with } \sigma = 0.1. \tag{33}
$$

where the mean vector $\mu(p_i) \in \mathbb{R}^d$ is designed as follows. First, we construct a pool of p_i values, where (i) 90% of which are drawn uniformly in the range $\log_{10}(p_i) \sim \mathcal{U}[-4,-3]$, and (ii) the remaining 10% are drawn uniformly in the range $\log_{10}(p_i) \sim \mathcal{U}[-6,-5]$. We then assign each p_i to a data point by sampling without replacement from this pool. Then, we define d quantile levels as follows

$$
\tau_j = 0.1 + 0.8\,\frac{j-1}{d-1} \quad j = 1, \ldots, d.
$$

Next, for a given probability p_i, we compute:

$$
\tilde{\mu}(p_i) = \left[F^{-1}_{\mathrm{Geom}(p_i)}(\tau_1)^{1/4}, \ldots, F^{-1}_{\mathrm{Geom}(p_i)}(\tau_d)^{1/4}\right]^\top,
$$

where $F^{-1}_{\mathrm{Geom}(p_i)}$ is the quantile function of the Geometric distribution with success probability p_i. The 1/4 power transformation reduces the size of the often extremely high raw quantiles. Lastly, to improve numerical stability, we normalize this vector by the average magnitude across all training examples:

$$\bar{\mu} = \frac{1}{n\,d}\sum_{i=1}^{n}\sum_{j=1}^{d} F^{-1}_{\mathrm{Geom}(p_i)}(\tau_j)^{1/4}, \quad \mu(p_i) = \frac{\tilde{\mu}(p_i)}{\bar{\mu}}.$$

Model Architecture and Training. To estimate the unsafe generation probabilities, we train a neural network with four hidden layers of size 32 each. We used ReLU for non-linearity and a sigmoid for output. We optimize the BCE loss (see Section E.3) using AdamW with learning rate 10^{-4}, weight decay 10^{-5}, batch size 100, for 10 epochs.

Calibration Method Parameters. In all experiments, unless stated otherwise, we used the prior quantile $\tau_{\mathrm{prior}} = 10^{-1/4}$, with trimming threshold M set so that $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right) = 10$.

E.1.1 Additional Experiments

To confirm that our findings on synthetic data do not depend on a particular choice of calibration/test split, we repeated the experiment from Section 6.2 for 20 independent splits of calibration and test sets. Specifically, in each trial, we fixed the original training set of 45,000 prompts, then randomly partitioned the remaining pool into a calibration set of size 45,000 and a test set of size 10,000. All other settings follow the experimental setup described in Section E.1.

We repeated the experiment for the following values of average sampling budget per prompt $B/|I_2|$: $10, 25, 50, 100, 200, 300, 600, 1200$. In all experiments, we used the same prior quantile $\tau_{\mathrm{prior}} = 10^{-1/4}$, with trimming threshold M set so that $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right) = 10$, as in Section 6.2.

Figure 3 presents the performance of the methods introduced in this work for 20 data splits. Across every budget level, the performances closely match those from our original fixed-split experiment from Figure 1 in Section 6.2. Next, we evaluate how the target coverage level affects the constructed LPBs and their informativeness by

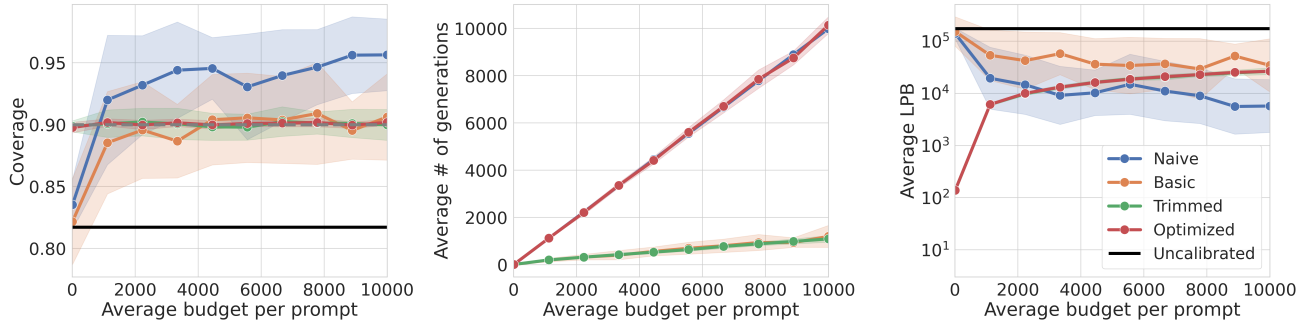

Figure 3: Results of synthetic experiments as a function of average budget per prompt $B/|\mathcal{I}_2|$. **Left:** Coverage (target 90%; gray dashed line). **Center:** Mean number of samplings generated per prompt. **Right:** Mean LPB. Shaded regions represent the standard deviation over 20 runs.

repeating the experiment from Section 6.2 across different nominal coverage levels. We present the results in Figure 4. This figure shows that the `Optimized` consistently achieves its target coverage across all levels.

Finally, we turn to study the effect of the maximal weight size $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right)$ on the stability and power of the `Trimmed` and `Optimized` calibration methods. Figure 5 compares the two methods under a fixed per-prompt budget of $B/|\mathcal{I}_2| = 1000$. For each weight size, we set the threshold $M = B\gamma/|\mathcal{I}_2|$. As can be seen in this figure, the size of the LPBs we construct increases with γ. However, this comes at the cost of increasing the coverage variance. Observe how the variance of `Optimized` method for $\gamma = 100$ is similar to that of `Trimmed` method for $\gamma = 10$, but the former yields much more informative LPBs under these corresponding γ values. This is a result of the variance-reducing optimization objective in (10).

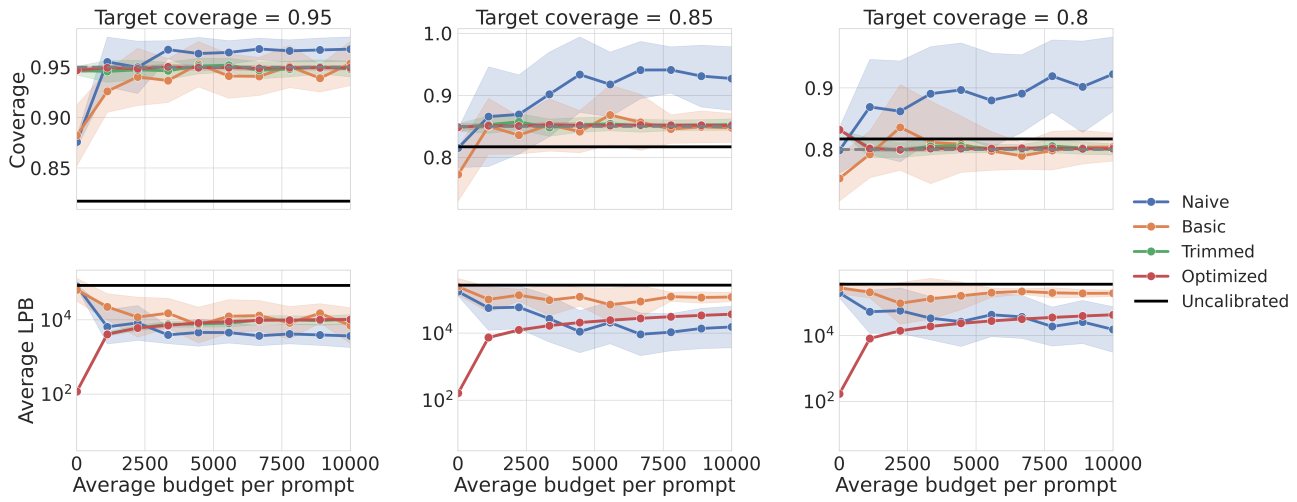

Figure 4: Results of synthetic experiments as a function of average budget per prompt $B/|\mathcal{I}_2|$. **Top:** Coverage (target level indicated by a gray dashed line). **Bottom:** Mean LPB. Shaded regions represent the standard deviation over 20 runs.

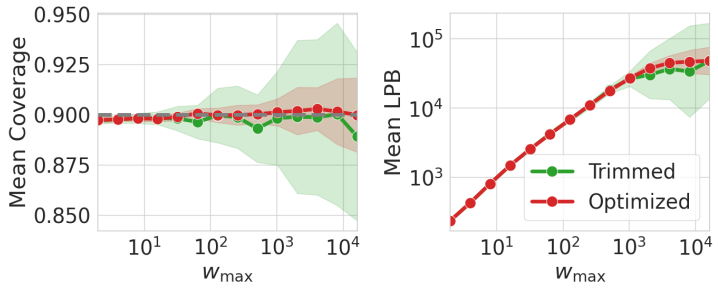

Figure 5: Results of synthetic experiments as a function of the maximum weight $w_{\max}$ of the `Trimmed` and `Optimized` methods. **Left**: Coverage (target 90%; gray dashed line). **Right**: Average LPB. Shaded regions present the standard deviation across 20 runs.

We compare the proposed `Optimized` method with an oracle quantile regression model on our synthetic dataset. The oracle model outputs the true conditional quantile of $T \mid X$ as the LPB, and therefore achieves the target coverage rate of $1-\alpha = 90\%$ by definition. These oracle LPBs are also the most informative possible by construction. Figure 6 reports the performance of the `Uncalibrated` quantile regression model, `Naive`, `Basic`, `Trimmed`, `Optimized`, and the oracle quantile regression model. As expected, the oracle model achieves the nominal coverage level. Notably, our `Optimized` method produces LPBs of comparable size to the oracle as the budget increases, demonstrating its efficiency in approaching the best achievable performance even without access to the oracle quantiles.

We compare our approach with a calibration method that has an infinite budget on our synthetic data. With an unlimited budget, we can observe an unsafe event for every calibration sample and thus can employ classic conformal prediction (Vovk et al., 2005; Papadopoulos, 2008) without data re-weighting to obtain LPBs that achieve $1-\alpha = 90\%$ coverage rate. Specifically, we estimate the miscoverage rate similarly to (6), but here, we use all calibration points:

$$\hat{\alpha}^{\text{ib}}(\tau) = \frac{1}{|\mathcal{I}_2|+1} \sum_{i \in \mathcal{I}_2} \mathbb{I}\{T_i < \hat{q}_\tau(X_i)\} \tag{34}$$

and construct the LPB as in (7) with the above $\hat{\alpha}^{\text{ib}}$.

We summarize the performance of all methods in Figure 7. The results reveal that the infinite budget method achieves the desired coverage rate with lower variability than the alternatives. Moreover, our `Optimized` ap-

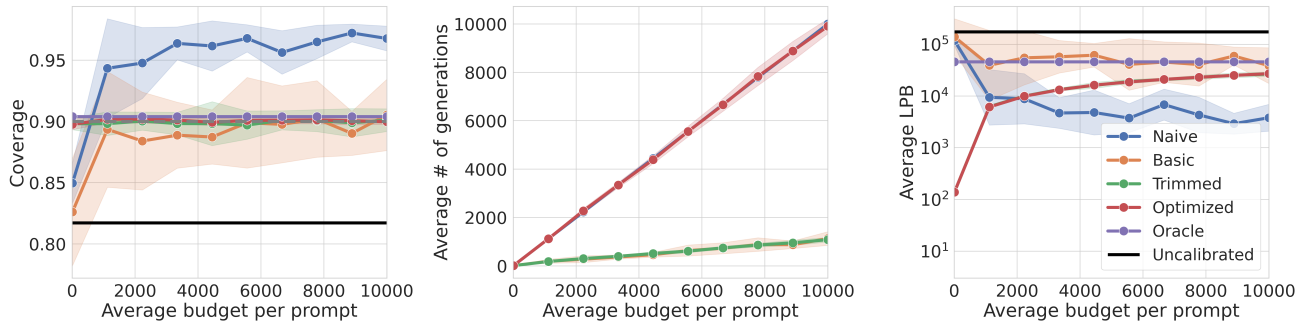

Figure 6: Synthetic data experiment with an oracle quantile regression model. **Top:** Coverage (target 90%%; gray dashed line). **Bottom:** Mean LPB. Shaded regions represent the standard deviation over 20 runs.

proach produces LPBs comparable in size to those of the infinite-budget calibration as the budget increases, demonstrating its effectiveness in approaching optimal performance even under budget constraints.

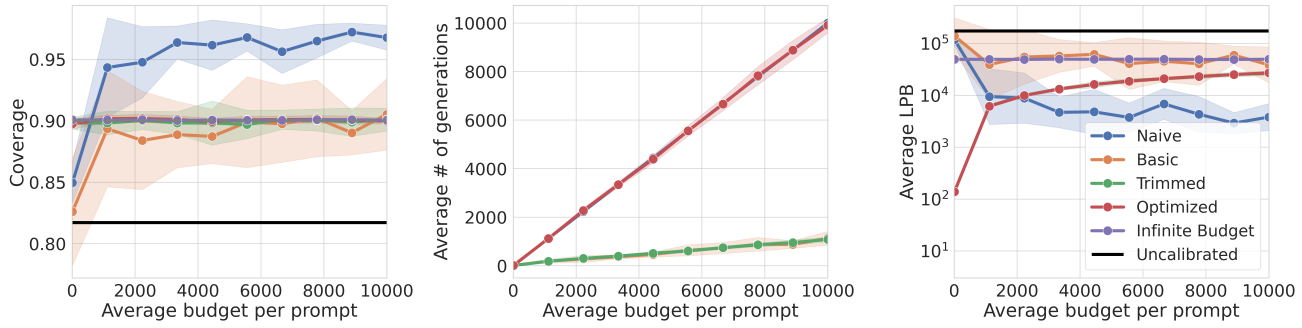

Figure 7: Synthetic data experiment with a calibration method that has an infinite budget. **Top:** Coverage (target 90%; gray dashed line). **Bottom:** Mean LPB. Shaded regions represent the standard deviation over 20 runs.

E.2 Real-Data Experiments

Calibration Method Parameters. In all experiments, unless stated otherwise, we used the prior quantile $\tau_{\text{prior}} = 10^{-1/4}$, with trimming threshold M set so that $\gamma = \max\left(|\mathcal{I}_2| \cdot M/B, 1\right) = 2$.

Efficient Prompt-Parallel Sampling. To efficiently generate outputs during our calibration procedures, we implement a prompt-parallel generation algorithm on top of the vLLM package (Kwon et al., 2023). In each iteration, we gather a batch of 1500 prompts X_i along with their corresponding censoring budgets C_i, and distribute them across multiple GPUs. The pipeline iteratively performs the following steps in parallel for a batch of prompts, until all prompts in the calibration set are processed:

1. Generate an output $\mathcal{G}(X_i)$ using the LLM.
2. Apply the audit function $\text{Audit}\big(X_i, \mathcal{G}(X_i)\big)$ to identify unsafe outputs.
3. Check whether X_i has either: (i) reached the C_i generations limit; or (ii) produced an output that fails the audit. If at least one of these two conditions is met, remove the prompt from the pipeline and replace it with a new prompt.

By promptly removing any prompt that yields an unsafe output, the processing maintains high throughput and avoids idle GPU time.

Runtime Evaluation. Table 1 presents average runtime for the `Naive` and `Optimized` calibration procedures across varying prompt budgets. While the `Naive` method can have a long runtime due to its unbounded censorship times C_i, it is observed to be faster than the `Optimized` method; see Table 1. We explain this by two key factors. First, our implementation of the `Naive` method includes an efficient runtime optimization: instead of always generating all C_i samples, we terminate generation early when additional samples no longer improve the estimate of $\hat{\alpha}(\tau)$. As detailed in Section E.3.1, this strategy yields LPBs that are equivalent to those produced by the original `Naive` procedure, while significantly reducing computation time. Second, the base model used in these experiments tends to produce low predicted quantiles $\hat{q}_\tau(X_i)$, as shown in Figure 2. Consequently, the effective censorship budgets C_i are also low for most prompts, further limiting runtime.

In sum, these two factors—runtime-efficient implementation and tendency to produce low predicted quantiles—explain why the reported runtime of the `Naive` is lower than the `Optimized` method in this setting.

Budget per prompt	Naive runtime (hours)	Optimized runtime (hours)	# GPUs
10	0.032 ± 0.005	0.063 ± 0.001	6
25	0.042 ± 0.001	0.117 ± 0.001	6
50	0.070 ± 0.006	0.209 ± 0.001	6
100	0.124 ± 0.003	0.388 ± 0.003	6
200	0.215 ± 0.010	0.589 ± 0.007	6
300	0.316 ± 0.012	0.718 ± 0.002	6
600	0.561 ± 0.016	0.979 ± 0.006	6
1200	0.918 ± 0.051	1.792 ± 0.041	4

Table 1: Average runtime (hours) by method and budget per prompt (mean ± SD).

Model Architecture and Training. We employ the ModernBERT-base (Warner et al., 2024) model. We use Adam optimizer (Kingma and Ba, 2015) to minimize the loss function in (35), with a learning rate of 3e-5, and a batch size of 1500. We train the model for 10 epochs and use early stopping based on the validation set.

E.3 Shared Implementation Details for Both Real and Synthetic Experiments

Our calibration framework applies to any machine learning model that outputs an estimated conditional quantile of the time-to-unsafe-sampling, denoted by $\hat{q}_\tau(x)$, given the covariates x. This is motivated by the fact that T, the time-to-unsafe-sampling, is a Geometric random variable with parameter $p(x)$, the probability of unsafe generation. Consequently, the conditional quantile function of T is given analytically in terms of $p(x)$, and vice versa:

$$q_\tau(x) = \left\lceil \frac{\log(1-\tau)}{\log(1-p(x))} \right\rceil \quad \text{and} \quad p(x) = 1 - (1-\tau)^{1/q_\tau(x)}.$$

The above relationship allows us to estimate either $p(x)$ or $q_\tau(x)$ and recover the other via a closed-form transformation.

Loss Function. In our experiments, we estimate $p(x)$ by fitting a model that minimizes the binary cross-entropy (BCE) loss over aggregate unsafe proportions. Specifically, for each prompt X_i, we define the empirical success rate as $\bar{Y}_i = \frac{1}{N}\sum_{j=1}^{N} Y_i^j$, where $Y_i^j \in \{0,1\}$ indicates whether the j-th sample is unsafe. We then use the BCE loss:

$$\text{BCE}(\bar{Y}_i, \hat{p}(X_i)) = -\left[\bar{Y}_i \log \hat{p}(X_i) + (1-\bar{Y}_i)\log(1-\hat{p}(X_i))\right], \tag{35}$$

where $\hat{p}(X_i)$ is the model's prediction of the probability of unsafe generation. Notably, this loss is equivalent to the mean of the standard BCE loss across individual samples:

$$\frac{1}{N}\sum_{j=1}^{N} \text{BCE}(Y_i^j, \hat{p}(X_i)) = \text{BCE}(\bar{Y}_i, \hat{p}(X_i)).$$

This aggregation allows for a more computationally and memory-efficient implementation compared to the use of the vanilla BCE.

E.3.1 Implementation Details of the `Naive` Calibration Procedure

In theory, the `Naive` calibration (Algorithm 4) can assign an unbounded censorship time per prompt, which leads to arbitrarily long running times. To avoid this, we restrict our search for the threshold τ to the compact set $\mathcal{T} \cap \tau_{\text{prior}}$ (see Section 6.3). This restriction in search space does not violate the method's validity, as it may only make it more conservative, as discussed in Section 6.1. This section also shows that assigning a censorship time $C_i \geq \hat{q}_\tau(X_i)$ is equivalent to drawing exactly $\hat{q}_\tau(X_i)$ generations. Conversely, setting $C_i < \hat{q}_\tau(X_i)$ amounts to not drawing any samples. Based on this observation, we implement a run-time efficient version of the `Naive` method that draws censorship times according to

$$C_i := \text{Ber}(g(X_i)) \cdot \hat{q}_{\tau_{\text{prior}}}(X_i),$$

where $g(x) = \mathbb{P}(\text{Geom}(\min(|\mathcal{I}_2|/B, 1) \geq \hat{q}_{\tau_{\text{prior}}}(X_i))$. It is important to note at this point that the average number of generations per prompt reported in Figure 1 is the simulated number of samples that the equivalent, non-optimized `Naive` solution would use. This implementation also implies that the `Naive` method has a much smaller overall number of generations, leading to a faster runtime.

E.4 Machine's Spec

The computational infrastructure used in the experiments are:

- **CPU**: Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz, Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz, Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz.
- **GPU**: NVIDIA A40, NVIDIA TITAN X (Pascal), NVIDIA 2080 TI, NVIDIA RTX 2060 SUPER.
- **OS**: Ubuntu 20.04.6.