

Mitigating LLM Hallucinations via Conformal Abstention

Yasin Abbasi-Yadkori*, Ilja Kuzborskij*, David Stutz, András György, Adam Fisch, Arnaud Doucet, Iuliya Beloshapka, Wei-Hung Weng, Yao-Yuan Yang, Csaba Szepesvári, Ali Taylan Cemgil, Nenad Tomasev

May 6, 2024

Abstract

We develop a principled procedure for determining when a large language model (LLM) should abstain from responding (e.g., by saying “I don’t know”) in a general domain, instead of resorting to possibly “hallucinating” a non-sensical or incorrect answer. Building on earlier approaches that use self-consistency as a more reliable measure of model confidence, we propose using the LLM itself to self-evaluate the similarity between each of its sampled responses for a given query. We then further leverage conformal prediction techniques to develop an abstention procedure that benefits from rigorous theoretical guarantees on the hallucination rate (error rate). Experimentally, our resulting conformal abstention method reliably bounds the hallucination rate on various closed-book, open-domain generative question answering datasets, while also maintaining a significantly less conservative abstention rate on a dataset with long responses (Temporal Sequences) compared to baselines using log-probability scores to quantify uncertainty, while achieving comparable performance on a dataset with short answers (TriviaQA). To evaluate the experiments automatically, one needs to determine if two responses are equivalent given a question. Following standard practice, we use a thresholded similarity function to determine if two responses match, but also provide a method for calibrating the threshold based on conformal prediction, with theoretical guarantees on the accuracy of the match prediction, which might be of independent interest.

1 Introduction

Large language models (LLMs) are excellent at next word prediction. At the same time, however, they are also prone to *hallucination*—that is, confidently generate responses that may look plausible on the surface, but that are actually incorrect or even nonsensical Ji et al. (2023); Maynez et al. (2020). Unfortunately, hallucinations are difficult to detect, especially when users are not able to easily verify the factuality of an LLM’s responses by themselves. In generation tasks in particular, it can be challenging to discriminate between hallucinations that present false facts, and any of the many other viable ways of expressing correct information. Therefore, hallucinations can be extremely detrimental towards achieving trustworthy and reliable LLM performance, and hence avoiding or even detecting hallucinations has become one of the most important research topics in LLM research.

In this work, we develop a principled abstention policy that mitigates LLM hallucination by simply choosing to either produce a single response from the model that is likely to be hallucination-free, or otherwise abstain from producing a response altogether (e.g., by saying “I don’t know”). The quality of such a policy can be measured by two quantities: the expected proportion of time the method chooses to abstain, and the expected proportion of unfiltered hallucinations in the responses; we will henceforth refer to these as the *abstention rate* and the hallucination *risk*, respectively.

*Equal contribution

While directly considering the (log-)probabilities of the response sequence generated by an LLM might be tempting, these probabilities heavily depend on the length of the output sequence, and the likelihood of an answer becomes non-indicative of its correctness as the sequence length grows (Manakul et al., 2023). Therefore, a large body of prior work has attempted to detect hallucinations through either confidence estimation (Cole et al., 2023; Manakul et al., 2023; Kuhn et al., 2023; Wang et al., 2022) or more involved inference time procedures. A consistent observation that has been reported in prior work is that *uncertainty* of the LLM responses, or equivalently, the level of *agreement* between a batch of sampled responses, tends to be a reasonable proxy for detecting hallucinations, although it clearly cannot detect situations where the LLM is completely sure about an incorrect answer. This approach comes with two immediate challenges: how we can decide if two responses agree for a given question, and what level of disagreement indicates hallucination.

In this paper we address both of these questions, by (i) developing well-engineered prompts to use the LLM for evaluating the similarity of two of its responses for a given query; and (ii) using theoretically well-founded methods to determine the level of agreement in evaluation responses, below which the LLM is likely hallucinating. A crucial property of (i) is that the *self*-evaluation prompt depends on the query itself, making it explicit that similarity of two responses depends on the question. For (ii), we leverage the *conformal prediction* and related risk control techniques (Vovk et al., 2005; Bates et al., 2021; Angelopoulos et al., 2021, 2024), by assuming access to a small holdout calibration set of prompt-response pairs. These techniques allow us to *calibrate* the detection/abstention policy so that it satisfies a pre-specified, distribution-free, statistical upper bound on the hallucination risk while minimizing the abstention rate. Our method is lightweight as it is only based on prompting and does not require to update the LLM itself, such as by fine-tuning.

We evaluate our method on a variety of closed-book open-domain question answering tasks (using a Gemini Pro model, Gemini Team, Google 2023). In particular, as also observed in parallel work (Kuhn et al., 2023; Manakul et al., 2023), we find that an instruction-tuned LLM can effectively and efficiently be used not only to generate candidate responses, but also to self-evaluate the coherence among responses; we then use the latter either to select a final response or to choose to abstain. We find that abstention with self-evaluation outperforms log-probability baselines used in the literature (Quach et al., 2023; Azaria and Mitchell, 2023).

To evaluate the experiments automatically, one needs to determine if two responses are equivalent given a question. A standard way to do this is to use a thresholded similarity function to determine if two answers match (Quach et al., 2023). To select the right threshold, we provide a calibration method, also based on conformal prediction, which comes with theoretical guarantees on the accuracy of the match prediction, and applicable for small calibration datasets (which need to be labelled manually). To our knowledge, this is the first such method presented in the literature, and hence it might be of independent interest.

2 Problem definition

We now give a formal definition of the problem we consider and summarize our approach. Let $\mathcal{X}$ be a space of input prompts and $\mathcal{Y}$ be a space of output responses. Let $m : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow \{0, 1\}$ be the binary ground-truth match function, so that $m(X; Y', Y) = 1$ indicates that response $Y' \in \mathcal{Y}$ matches the response $Y \in \mathcal{Y}$ for a given query $X \in \mathcal{X}$, and $m(X; Y', Y) = 0$ denotes that it does not. That is, given a ground truth response Y to X , $m(X; Y', Y)$ is the indicator function whether Y' is semantically equivalent to Y *given* X . The conditioning on X makes our model very flexible: While the simplest way to define m could be to check if Y and Y' mean the same thing, our setting can accommodate much broader and more useful definitions, the most appealing of which is whether Y and Y' are equally correct responses to X . For example, for the prompt $X = \text{"Tell me a European capital."}$, $Y = \text{"London"}$ is as good as $Y' = \text{"Paris"}$, allowing our method to be applicable for questions with multiple different correct responses, as long as a good match function m can be devised.

Given a classifier (i.e., a possibly random map) $f : \mathcal{X} \rightarrow \mathcal{Y}$, its loss on the prompt-response pair (X, Y) is defined as $1 - m(X; f(X), Y)$. Our goal is to obtain, given a classifier f , a selective classification scheme which can abstain from prediction (answering a prompt) when f would make a mistake. To this end, we define an *abstention* function, which can decide whether the classifier should be applied to a given input prompt X . We consider score-base abstention functions, that is, for a given parameter $\lambda \in \Lambda$ (where $\Lambda \subset \mathbb{R}$

is a parameter space), a query $X \in \mathcal{X}$, and a score function $g : \mathcal{X} \rightarrow \mathbb{R}$ indicating the model’s confidence in classifying the input, the abstention policy $a : \Lambda \times \mathcal{X} \rightarrow \{0, 1\}$ is defined as

$$a_\lambda(X) = \begin{cases} 1 & \text{if } g(X) < \lambda \\ 0 & \text{if } g(X) \geq \lambda \end{cases}$$

where $a_\lambda(X) = 1$ means that the predictor should abstain. Given a query X , the score might be a random variable, and therefore a , similarly to f , might also be random. Together the pair (a_λ, f) define a *selective classifier*.

Let $\ell : \mathcal{X} \times \mathcal{Y} \times \Lambda \rightarrow \mathbb{R}$ be a loss function so that $\ell(X, Y; \lambda)$ is the loss of selective classifier (a_λ, f) given a query-response pair (X, Y) . ℓ penalizes a policy when it does not abstain and its response does not match the label:

$$\ell(X, Y; \lambda) = (1 - a_\lambda(X))(1 - m(X; f(X), Y)). \quad (1)$$

A trivial policy that always abstains would result in a zero loss. However, an interesting policy would also have a small abstention rate. The quality of a policy that can abstain is controlled by: (i) the *risk* $R(\lambda) = \mathbb{E}[\ell(X, Y; \lambda)]$ of producing an incorrect answer on a new query, and (ii) the *rate of abstention* $T(\lambda) = \mathbb{E}[a_\lambda(X)]$, where the expectations are taken over a query-response pair (X, Y) distributed according to $\mathcal{D}$.

To balance these quantities, we are interested in finding the abstention threshold λ resulting in the smallest number of abstentions for a given risk tolerance $\alpha > 0$:

$$\operatorname{argmin}_{\lambda \in \Lambda} T(\lambda), \quad \text{subject to} \quad R(\lambda) \leq \alpha. \quad (2)$$

Since the abstention rate $T(\lambda)$ is a non-decreasing function of λ , this is equivalent to finding the smallest λ for which $R(\lambda) \leq \alpha$; we denote this optimal threshold by λ^* .

To solve this problem approximately, we assume that we are given a calibration dataset

$$D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \subset \mathcal{X} \times \mathcal{Y},$$

which is a collection of ground truth query-response pairs. We also assume that given a new test point (X, Y) sampled from the true data distribution $\mathcal{D}$, and that $\{(X, Y), (X_1, Y_1), \dots, (X_n, Y_n)\}$ are exchangeable¹ (which is a generalization of the assumption that they were all selected independently from $\mathcal{D}$). We will use the calibration dataset D_n to design our abstention policy, that is, to find a $\hat{\lambda}$ such that we can guarantee $R(\hat{\lambda}) \leq \alpha$ with high probability, based on D_n . Notice that the calibration dataset is much smaller than the training dataset that is used to train the LLM. Before discussing how λ is optimized (which is presented in Section 3), we first discuss potential choices for the classifier f and the score function g in our context.

2.1 Choice of the score function g and the classifier f

In this section, we discuss the choice of the score function g and the classifier f . Let k be an integer. We augment each question-answer pair (X_i, Y_i) with k samples $Y_i^1, \dots, Y_i^k$ generated from the LLM given a query X . So a datapoint in the calibration data will be of the form $(X_i, Y_i, Y_i^1, \dots, Y_i^k)$. Notice that in many use cases of LLMs, we already generate multiple responses for a given query and output a response based on a number of criteria. So we are not adding a computational overhead here by demanding the existence of k responses. We can choose k to be any number of responses the LLM already generates.

We consider two score functions. The first, called *match count*, is defined with respect to a contextual similarity function $s : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ (that might be different than the match function m) and is parameterized by a positive scalar parameter β . By default, we suggest using LLM prompting to measure similarity of text outputs, but other similarity functions could also be used. For a query X , generated responses $Y^1, \dots, Y^k$,

¹Jointly distributed random variables $Z_1, \dots, Z_n$ are exchangeable if for every permutation π of $[n]$, $P(Z_1, \dots, Z_n) = P(Z_{\pi_1}, \dots, Z_{\pi_n})$.

and a parameter β , let the score of response Y^i be the number of other responses that are similar to Y^i , that is, $|\{j \neq i : s(X; Y^i, Y^j) > \beta\}|$. The default response $f(X)$ is a response with the largest score, and the score $g(X)$ is the score of $f(X)$. As explained in the previous section, the policy abstains if the score is below λ . Otherwise, the policy returns the response $f(X)$.

Similarly as reported in the literature (Manakul et al., 2023; Kuhn et al., 2023), we have observed that using LLM prompting as the similarity function works well in practice. Computing the score function then requires $O(k^2)$ extra inferences, which adds significant computational overhead. There are multiple cheaper alternatives. One cheaper alternative is to get similarity of each response with all other responses in a single prompt. This alternative still performs well in practice while being much faster to compute. An even more interesting alternative, called *expected match count*, is the following: for each response Y^i , ask the LLM in a single query how many matches exist among other responses $\{j \neq i : Y^j\}$. Then the score is the expected match count $g(X) = \sum_{i=1}^k q(\text{"i"} \mid X) i$, where q is the probability of token "i" according to the LLM. In addition to being computationally inexpensive, this score can take any values in interval $[0, k]$, which allows for a more fine-grained and improved optimization. On the other hand, computing this score requires access to the log-probabilities of the LLM, and is not a black-box solution.

Finally, the simplest alternative is to choose f to be the greedy (zero-temperature) output of the LLM (denoted, say, by Y^1), and the score of this prediction is the number of similar responses in the randomly selected samples $Y^2, \dots, Y^k$, as defined either by the match count or the expected match count above. This approach reduces the computation cost of the comparisons by a factor of k , and we refer to it as the *greedy* version of the methods.

2.2 Choice of the match function m

Match functions can be naturally derived from similarity scores: two responses match if their similarity score is large enough (i.e., larger than a given threshold). A popular similarity score function, usually defined in term of a response and a true label, is the F1 score (Joshi et al., 2017; Devlin et al., 2019), which is calculated as

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

where precision is the percentage of the response words that appear in the label sentence, and recall is the percentage of the label words that appear in the response sentence. When the labels are short sentences, as is the case in our experiments, we can obtain more reliable results using only the recall score. For experiments on the TriviaQA dataset (Joshi et al., 2017), with short answers and labels, we use the recall score to evaluate different methods.

Both the F1 and the recall scores however are poor choices when LLM answers are longer and can be expressed in many forms. For experiments conducted on the Temporal Sequences dataset Srivastava et al. (2023) which we consider in the following, responses are sometimes long texts, and so we use LLM-prompting to decide if the generated answer and the label match, using the same similarity metric as before, by asking the LLM to measure similarity of two texts given the question on a scale of 1 – 10. If the score is above a pre-specified threshold, the generated text is considered correct (or a match). The same conformal risk control procedure (discussed in the next section) can be used to verify the validity of this match function choice. The details of the calibration of the match function are presented together with the descriptions of the experiments in Section 7.

In the next section, we discuss tuning of the abstention policy and the match function based on the calibration set.

3 Conformal abstention

Given the calibration dataset, we want to construct a postprocessing procedure that guarantees that the resulting composite policy (which depends on the calibration data, and hence, is random) is an approximately optimal solution for problem (2).

Notice that the loss function is non-increasing in λ : for $\lambda_1 \leq \lambda_2$, if $a_{\lambda_1}(X) = 1$, then $a_{\lambda_2}(X) = 1$ and both parameters have zero loss. On the other hand, f does not depend on λ , and hence the loss of λ_2 is smaller than or equal to the loss of λ_1 . Given that calibration data and the test point are exchangeable while the loss function ℓ is non-increasing in λ , then we can use the *Conformal Risk Control (CRC) framework* of Angelopoulos et al. (2024) to tune λ . In particular, define the average loss

$$L_n(\lambda) = \frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i; \lambda)$$

and let

$$\hat{\lambda}_n = \inf \left\{ \lambda : \frac{n}{n+1} L_n(\lambda) + \frac{1}{n+1} \leq \alpha \right\} . \quad (3)$$

Then, it holds that (Angelopoulos et al., 2024)

$$\mathbb{E}[R(\hat{\lambda}_n)] = \mathbb{E}[\ell(X, Y; \hat{\lambda}_n)] \leq \alpha . \quad (4)$$

The expectation in (4) is over calibration data as well as the test point. For completeness, the proof is given in Appendix A.

The above guarantee is non-trivial: standard confidence-interval-based methods would lead to a solution that uses a more conservative padding of order $O(1/\sqrt{n})$ instead of the smaller $O(1/n)$ padding used in (3). We will discuss this alternative approach, called Risk-Controlling Prediction Sets (RCPS), in Section 4.

3.1 Simple high probability amplification of the CRC procedure

The CRC formulation of Angelopoulos et al. (2024) given by (3) and (4) holds only in expectation over the calibration data. To have reliable decision making, in practice one typically desires to have confidence guarantees that hold with high probability over the samples. In this and the following section we present several methods that come with high-probability guarantees; some of these approaches will be compared experimentally in Section 7.

Clearly, under the assumption that the loss function is non-negative, one can convert a CRC guarantee in expectation to a guarantee in probability, for instance through Markov's inequality. Namely, (3) and (4) imply that $\mathbb{P}(R(\hat{\lambda}) \geq \alpha/\delta) \leq \delta$ for any failure probability δ . This result is rather weak as it does not hold with high probability. However, interestingly, we can further improve upon Markov's inequality without extra assumptions through an amplification (or boosting) argument, at the expense of data splitting. Unlike Markov's inequality, such inequality provides a high-probability guarantee for the CRC procedure, however it is looser by a constant factor than the high-probability inequalities we will consider for RCPS in the coming section.

Proposition 3.1. *Assume that loss function $\lambda \mapsto \ell(z; \lambda)$ is non-increasing, right-continuous, and bounded within $[0, B]$ for any z . Assume data D_n is composed of i.i.d. samples. Let the failure probability be $\delta \in (0, 1)$ and the desired error be $\alpha > 0$. Then, consider an arbitrary partition $D_n = (S_1, \dots, S_K)$ where $K = \lceil \ln(1/\delta) \rceil$, and let*

$$\hat{\lambda}_i = \inf \left\{ \lambda \in \Lambda : \frac{n}{n + \ln(\frac{1}{\delta})} L(S_i; \lambda) + \frac{B \ln(\frac{1}{\delta})}{n + \ln(\frac{1}{\delta})} \leq \alpha \right\} .$$

Then, for $\hat{\lambda}^* = \max_{i \in [K]} \hat{\lambda}_i$,

$$\mathbb{P} \left(R(\hat{\lambda}^*) \leq e \alpha \right) \geq 1 - \delta .$$

In particular, the constant e can be replaced by $c > 1$ while also replacing $\ln(\frac{1}{\delta})$ by $\frac{\ln(1/\delta)}{\ln(c)}$.

Proof of Proposition 3.1. The proof is based on (3) and (4). In particular, for some $\gamma > 0$, consider the following probability:

$$\mathbb{P} \left(\bigwedge_{i=1}^K R(\hat{\lambda}_i) > \frac{\mathbb{E}[R(\hat{\lambda}_i)]}{\gamma} \right) = \prod_{i=1}^K \mathbb{P} \left(R(\hat{\lambda}_i) > \frac{\mathbb{E}[R(\hat{\lambda}_i)]}{\gamma} \right) \leq \gamma^K,$$

where we note that the probability factorizes since $(\hat{\lambda}_i)_i$ are fitted using independent samples and the last inequality follows from Markov's inequality. Now, choosing $\gamma = 1/e$, we have that

$$\mathbb{P} \left(\exists i \in [K] \quad R(\hat{\lambda}_i) \leq e \mathbb{E}[R(\hat{\lambda}_i)] \right) \geq 1 - e^{-K} \implies \mathbb{P} \left(\exists i \in [K] \quad R(\hat{\lambda}_i) \leq e \alpha \right) \geq 1 - e^{-K}$$

where the last step follows by (4) with $S' = S_i$ and so $|S'| = n/K = n/\lceil \ln(1/\delta) \rceil$.

Finally, since our result so far only guarantees the existence of one $\hat{\lambda}_i$ that succeeds with high probability, we use the property that $L(\cdot)$ is non-decreasing to claim that

$$\mathbb{P} \left(R(\max_{i \in [K]} \hat{\lambda}_i) \leq e \alpha \right) = \mathbb{P} \left(\min_{i \in [K]} R(\hat{\lambda}_i) \leq e \alpha \right) \geq \mathbb{P} \left(\exists i \in [K] \quad R(\hat{\lambda}_i) \leq e \alpha \right).$$

The statement then follows. $\square$

3.2 Bounding $|\lambda^* - \hat{\lambda}_n|$

Although the CRC procedure ensures that the constraint inequality in (2) is satisfied, it provides no guarantees on how the abstention rate of the solution $\hat{\lambda}_n$ deviates from the optimal abstention rate, i.e. a bound on $|T(\lambda^*) - T(\hat{\lambda}_n)|$.

Assuming that the abstention rate T and risk R are differentiable functions of the threshold λ , under the assumptions of Theorem 2 of Angelopoulos et al. (2024), we have

$$\mathbb{E}[R(\hat{\lambda})] \geq \alpha - \frac{2}{n+1} \geq R(\lambda^*) - \frac{2}{n+1} \implies R(\lambda^*) - \mathbb{E}[R(\hat{\lambda})] \leq \frac{2}{n+1}.$$

Let $\gamma = \sup_{\lambda} \frac{dT(\lambda)}{dR(\lambda)}$ and assume it is finite. Then,

$$\begin{aligned} T(\lambda^*) - \mathbb{E}[T(\hat{\lambda})] &= \mathbb{E}[T(\lambda^*) - T(\hat{\lambda})] \\ &\leq \mathbb{E} \left[\gamma (R(\lambda^*) - R(\hat{\lambda})) \right] \\ &\leq \frac{2}{n+1} \gamma. \end{aligned}$$

4 Risk-Controlling Prediction Sets

Motivated by the need for high-probability guarantees over the calibration data, Angelopoulos et al. (2021) also introduced another family of methods, called RCPS. Indeed, for a loss bounded by one as we have, we can get for $\hat{\lambda}_n$ obtained using conformal risk control the following high-probability result:

$$\mathbb{P} \left(\mathbb{E}[\ell(X, Y; \hat{\lambda}_n) \mid D_n] \leq \alpha + c(\delta, \alpha, n) \right) \geq 1 - \delta, \quad (5)$$

for $c(\delta, \alpha, n) = u(\delta, n) - v(\alpha, n)$, where the probability is over the calibration set D_n and

$$u(\delta, n) = \sqrt{\frac{\log(\frac{1}{\delta})}{2n}} \quad \text{and} \quad v(\alpha, n) = \frac{1 - \alpha}{n}.$$

This follows from the fact that $\hat{\lambda}_n$ can be rewritten as

$$\begin{aligned}\hat{\lambda}_n &= \inf \left\{ \lambda : \frac{n}{n+1} L_n(\lambda) + \frac{1}{n+1} \leq \alpha \right\} \\ &= \inf \left\{ \lambda : L_n(\lambda) + v(\alpha, n) \leq \alpha \right\} \\ &= \inf \left\{ \lambda : L_n(\lambda) + u(\delta, n) \leq \alpha + u(\delta, n) - v(\alpha, n) \right\} \\ &= \inf \left\{ \lambda : L_n(\lambda) + u(\delta, n) \leq \alpha + c(\delta, \alpha, n) \right\}\end{aligned}$$

This identity shows that applying conformal risk control is equivalent to applying the distribution-free RCPS procedure of Bates et al. (2021) for a Hoeffding upper confidence bound (UCB) on the empirical risk at level $\alpha + c(\delta, \alpha, n)$. It then follows from Theorem 2 of Bates et al. (2021) that (5) holds.

However, the RCPS approach is more general than CRC as it is applicable even if the loss function is non-monotonic.² Let $\delta \in (0, 1)$ be a failure probability. We want to choose $\hat{\lambda}$ to ensure that

$$\mathbb{P}(\mathbb{E}[\ell(X, Y; \hat{\lambda}) | D_n] \leq \hat{R}_{\text{ub}}(\hat{\lambda})) \geq 1 - \delta.$$

Here, $\mathbb{P}$ is again over the random calibration set D_n . In the following we consider several upper confidence bounds for RCPS, some of which were already discussed by Bates et al. (2021).

Baseline confidence bounds Among RCPS methods we first consider the *empirical Bernstein inequality* (Audibert et al., 2007; Maurer and Pontil, 2009). In this case, the upper bound is computed as

$$\hat{R}_{\text{ub-bern}}(\lambda) = L_n(\lambda) + \sqrt{\frac{2\widehat{\text{Var}}(\lambda) \ln(\frac{2}{\delta})}{n}} + \frac{7}{3} \frac{\ln(\frac{2}{\delta})}{2(n-1)}$$

where $\widehat{\text{Var}}(\lambda)$ is a sample variance of losses computed with parameter λ .

Since we are working with Bernoulli losses, we evaluate the *Hoeffding-Bentkus inequality*, one of the tightest known bounds for such losses. Computation of the bound relies on the following function (of $t, p \in [0, 1]$),

$$\varepsilon_{\text{hb}}(t, p) = \min \left\{ e^{-n \text{kl}(t, p)}, \mathbb{P}(\text{Bin}(n, p) \leq \lceil nt \rceil) \right\},$$

where $\text{Bin}(n, p)$ is a binomial random variable with parameters $n \in \mathbb{N}$ and $p \in [0, 1]$ and $\text{kl}(p, q) = p \ln(p/q) + (1-p) \ln((1-p)/(1-q))$ is the relative entropy between two Bernoulli distributions with success probabilities p and q , respectively. Then, the upper bound is given by solving a simple optimization problem

$$\hat{R}_{\text{ub-hb}}(\lambda) = \sup \{ p \in [0, 1] : \varepsilon_{\text{hb}}(L_n(\lambda), p) \geq \delta \}.$$

Finally, we consider the so-called *Bernoulli relative-entropy inequality*, a.k.a. the ‘little kl’ inequality (see, for instance, Maurer, 2004). Here the upper confidence bound is computed by solving the simple optimization problem

$$\hat{R}_{\text{ub-kl}}(\lambda) = \sup \left\{ p \in [0, 1] : \text{kl}(L_n(\lambda), p) \leq \frac{\ln(\frac{\sqrt{n}}{\delta})}{n} \right\}.$$

Bates et al. (2021) mentions another, so-called Waudby-Smith-Ramdas (WSR) inequality for the case of non-binary losses, which is tighter for such cases since it adapts better to the variance. This inequality belongs to the family of concentration inequalities derived through regret analysis of online betting algorithms, first proposed by Kwang-Sung and Orabona (2019). In fact, it was recently shown that WSR inequality is looser than another inequality from this family (Orabona and Jun, 2023), and which notably, for Bernoulli distributions coincides with the Bernoulli relative-entropy inequality considered above.

²When the loss function is non-monotonic we can no longer rely on arguments as in (5), however we can still apply confidence bounds discussed here by making them hold uniformly over a finite parameter set Λ through the union bound argument. In such case, δ is replaced by $\delta/|\Lambda|$.

5 Calibrating the match function m

As described in Section 2.2, it is hard to identify if two responses to a query (e.g., one generated by the LLM and another being the ground truth answer) are the same, and hence devising a good match function (to be used in computing the loss ℓ) is a non-trivial problem. As explained before, we consider score-based match functions. Then, given a similarity score $s : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, it is natural to define m as a thresholded version of the score function:

$$m(X, Y', Y) = m_\beta(X, Y', Y) = \mathbb{I}\{s(X, Y, Y') \geq \beta\}$$

where for any event E , $\mathbb{I}\{E\}$ denotes its indicator function and β is a threshold to be chosen. Note that although we use the same notation for the similarity function and its threshold as in the definition of the score function in Section 2.1, these are not necessarily the same.

In this section we assume that s is given (in the experiments we will use different option, such as recall or LLM self-prompting, discussed in Section 2.2), and the goal is to select a threshold β so that the match function m reflects the ground truth as much as possible (given s). We can do this based on another calibration set, again, with a slight inconsistency in the notation, denoted by $(X_1, Y'_1, Y_1), \dots, (X_n, Y'_n, Y_n)$, where, for all i , (X_i, Y_i) are ground-truth question-answer pairs sampled independently from the data distribution $\mathcal{D}$, and Y'_i is the model's response to query X_i . Whether Y_i and Y'_i agree has to be checked manually, so the size n of this calibration set can be quite small in practice.

If the quality of the responses is monotone in s , that is, if $s(X, Y', Y) < s(X, Y'', Y)$ means that Y'' is a better response to X than Y' (as suggested by the ground truth response Y), then one can use any of the methods discussed in the previous sections, such as (3), to select a threshold β to get a guarantee on the error the match function makes when comparing responses to the ground truth; here we can define $\ell(X_i, Y'_i, Y_i)$ to be 0 if the match function is correct about comparing Y_i and Y'_i and 1 otherwise.

However, none of our similarity function candidates are monotone, as typically a too high threshold becomes too conservatives and may classify some correct responses Y'_i as incorrect, while a too low threshold may result in incorrect answer classified as correct. Nevertheless, we present next a procedure which, using an upper bound on the performance of m , allows us to calibrate the threshold β with theoretical guaranties.

Let C be the number of incorrect LLM responses (i.e., when the LLM's response does not match the label according to the human rater) for our n calibration samples. This is the true performance measure. Let L_2 denote the number of times the LLM's response is different from the label according to the match function m . This is the performance measure that we report when we use m as a surrogate to the true loss. Next we show how proper calibration of β can ensure that L_2 is an approximate upper bound on C , and hence reporting errors based on m can be used to upper bound the true error rate. Let L_1 denote the number of times the LLM's response is different from the corresponding label, but is classified as the same according to the match function. Then clearly

$$C \leq L_1 + L_2.$$

While the dependence of L_2 on the threshold β can be arbitrary in general, it is easy to see that L_1 is a monotone decreasing function of β (setting a higher threshold β either keeps $m(X, Y, Y')$ unchanged or changes it from 1 to 0), allowing the application of conformal prediction to set β with theoretical guarantees on the behavior of L_1 on new data using the calibration dataset $(X_1, Y'_1, Y_1), \dots, (X_n, Y'_n, Y_n)$.

For example, setting the value of β according to the conformal prediction rule (3) as

$$\hat{\beta} = \inf \left\{ \beta : \frac{n}{n+1} \sum_{i=1}^n (1 - m_\beta(X_i, Y'_i, Y_i)) + \frac{1}{n+1} \leq \alpha \right\}$$

guarantees that

$$\mathbb{E}[L_1] \leq \alpha$$

on new test data, which implies that on expectation $L_2 + \alpha$ is an upper bound on the number of errors the LLM makes. Note that since L_2 is evaluated using the (calibrated) match function m , at test time we can

use a lot of data, making the measured value of L_2 arbitrarily close to its expectation $\mathbb{E}[L_2]$ (we can also use any of the confidence bounds from Section 4 to upper bound their difference), we can guarantee that the expected number of errors $\mathbb{E}[C]$ made by the LLM satisfies

$$\mathbb{E}[C] \leq \mathbb{E}[L_2] + \alpha \leq L_2 + \alpha + \epsilon, \quad (6)$$

where $\epsilon \geq \mathbb{E}[L_2] - L_2$ is an upper bound on the difference of L_2 and its expectation, which can be made arbitrarily small. Selecting a calibration method which comes with a high-probability guarantee would yield a high-probability version of (6).

6 Related work

Uncertainty quantification of machine learning methods is a large and active area of research. We only discuss prior papers that are closely related to our work.

6.1 Selective classification

The problem that we study is a case of selective classification (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017; Lin et al., 2022). Given a classifier, a training set, a confidence parameter, and a desired risk bound, the objective of Geifman and El-Yaniv (2017) is to design an abstention policy such that the risk is bounded by the desired bound with high probability. They normalize loss by decision rate (one minus abstention rate), which makes loss non-monotonic. Geifman and El-Yaniv (2017) propose a binary search procedure. However, given the non-monotonicity of the loss function, the binary search procedure is not guaranteed to find a solution that satisfies the risk condition.

Kamath et al. (2020) study selective question answering when the test point might be out-of-domain. Selective classification methods are closely related to the RCPS approach that we discussed in Section 4.

6.2 Abstention in LLMs

There has been a number of recent papers that study abstention in LLMs. We only cover approaches that use a pre-trained model, and not those based on fine-tuning LLMs. These papers usually consider general metrics for their methods, such as the area under the curve, and do not provide any practical guidance on how to actually choose an abstention policy, which is one of our main contributions. Given a risk tolerance α , the policy that these papers implicitly suggest chooses a policy parameter that leads to α loss; while this method comes with no theoretical guarantees, we consider it as a baseline for our calibration methods in our experiments.

Cole et al. (2023) investigate a number of score functions in designing an abstention mechanism: (i) a likelihood-based score; (ii) using sampling repetition and counting how many times the sampled output matches exactly (after making the response lower case and removing punctuation) the greedy (zero-temperature) output; (iii) computing sampling diversity defined as the fraction of non-unique answers; and (iv) using self-verification by checking the probability given by the model if the greedy answer is correct. They report that their approach (ii) is generally the best. However, since it considers exact match of the responses, its applicability is limited to short responses only (otherwise exact matching almost never happens in practice). Furthermore, the resulting abstention policy has no theoretical (statistical) performance guarantee, unlike the one we propose here.

Manakul et al. (2023) study a black-box approach to detecting hallucinations by generating multiple responses, and measuring similarity of a reference response and the set of generated responses. They consider various measures of similarity, including LLM self-prompting. In their experiments, the method that generates multiple responses and uses LLM self-prompting for similarity calculations outperforms other baselines including the one using log-probability scores. Although the overall approach in this paper is conceptually similar to ours, their self-prompting method only compares responses without contexts, resulting in inferior

similarity measures. Similar methods have been studied by Lin et al. (2023). Furthermore, as discussed at the beginning of this section, the choice of an actual abstention policy is not discussed in either of these papers.

Kuhn et al. (2023) study uncertainty quantification of LLMs. Similarly to our work, they propose generating k responses, and clustering them based on their contextual similarities evaluated using a smaller language model. Then they investigate the application of semantic entropy to score model uncertainty.³ As entropy measures the uncertainty of the whole output distribution, this method does not seem to be directly applicable to decide between using a given response (e.g., the zero-shot response) or abstain.

Wang et al. (2022) study reasoning with LLMs and propose generating a set of ‘reasoning paths’ instead of a final answer. Here reasoning paths are generated by prompting the model to provide intermediate reasoning steps used to arrive at the answer. Instead of greedily choosing the ‘best’ answer according to some criterion of the associated reasoning path, the paper proposes to select the most consistent answer. This approach is complementary to the one we consider in this paper, and in principle, the match function and the score function can be computed using reasoning paths.

6.3 Using token probabilities to quantify uncertainty

A popular approach to quantify uncertainty is based on using (normalized) log-probabilities of responses. Kadavath et al. (2022) show that LLMs are well-calibrated at the token level on multiple-choice question-answering tasks when the prompts are in an appropriate format. However, the quality of log-probability scores quickly degrades as the model generates longer texts (Cole et al., 2023; Manakul et al., 2023; Kuhn et al., 2023).

6.4 Asking language models to quantify uncertainty (self-verification)

Kadavath et al. (2022) propose using LLM self-prompting to measure a model’s uncertainty in its responses. More specifically, for a given query, a number of responses are generated, and then the model is queried if the responses are correct. For this query, the log-probability of “True” is returned as a measure of uncertainty. Related approaches are studied by Mielke et al. (2022). However, Manakul et al. (2023) and Kuhn et al. (2023) report that LLM self-verification is not as effective as sampling-based methods (i.e., methods using multiple responses) in quantifying model uncertainty.

6.5 Applications of conformal prediction in quantifying uncertainty in LLMs

Conformal prediction has been used for quantifying uncertainty in LLMs, but we are not aware of any works that employ conformal prediction in designing an abstention mechanism. Quach et al. (2023) use conformal prediction to construct confidence sets of text outputs that contain an acceptable answer with a high probability, based on a calibration mechanism applied to log-probability scores. Ravfogel et al. (2023) propose using conformal prediction to calibrate parameter p in nucleus (top- p) sampling. Ren et al. (2023) consider a multiple-choice-style LLM planning, and use conformal prediction to quantify uncertainty of LLM-based planners.

6.6 Other uncertainty-quantification methods in deep learning and LLMs

Ensemble methods are based on the classical idea of bootstrap for confidence estimation (Tibshirani and Efron, 1993) where multiple estimators for the regression function, each computed on a perturbed version of the data (e.g. by drawing samples from the empirical distribution over data), are combined.

The empirical distribution of the resulting estimates is then used to construct confidence intervals. While many of these methods can be interpreted as sample-based approximations to Bayesian methods, model-hyperparameter selection (e.g., scale of perturbations, learning) for ensemble methods is typically done using a validation on holdout data (a subset of the training data). Many recent papers have studied ensemble

³Note that their formula (4) for estimating semantic entropy is incorrect, as it gives uniform weight to all clusters.

methods in the context of deep learning and reinforcement learning (Osband et al., 2016; Lakshminarayanan et al., 2017; Malinin and Gales, 2020). In the context of LLMs, the methods require training multiple language models, which is very expensive. Osband et al. (2023) introduces epistemic neural networks (epinets), which approximate ensemble methods by training a single network with an artificially injected (controlled) source of randomness. Rabanser et al. (2022) proposes to use intermediate model checkpoints to quantify the uncertainty of the final model in its responses. While these approaches aim to mimic the bootstrap procedure during prediction, their validity is not justified by theoretical considerations, and hence remain heuristic approximations.

7 Experiments

Our experiments aim to verify three hypotheses: (i) conformal abstention done through CRC and RCPS is able to mitigate hallucinations as measured by loss (1), while maintaining a low abstention rate; (ii) the loss (1) is a reasonable measure of detecting hallucinations; and (iii) for longer responses, defining scores using LLM similarity prompting is more effective than the ones based on log-probabilities.

Datasets. We evaluate our approach on two publicly available question-answering datasets: Temporal Sequences (a dataset from the BIG-bench benchmark of Srivastava et al., 2023) and TriviaQA (Joshi et al., 2017). TriviaQA predominantly contains short answers while Temporal Sequences contains several long answers as well. We hypothesise that some commonly used scores, such as log-probabilities predicted by the model will not yield a good performance on long answers, and therefore we expect that the calibration procedure combined with log-probability scores will perform worse than the calibration procedure with our proposed match-scores on Temporal Sequences.

Calibration/test splits. In each experiment, 20% of the data is used for testing (holdout sample). Each experiment is performed on subsamples of calibration sets of increasing sizes; moreover, each subsample is drawn with replacement 10 times. We report the resulting average test losses and their standard deviations. We also report the median for the parameter λ of our methods.

Language model. We use a Gemini Pro model (Gemini Team, Google, 2023) to generate outputs and scores.

The match function m . We use a similarity-score-based match functions to compute the loss ℓ , as described in Section 2.2, and calibrate its threshold according to Section 5. Thus, first we have to choose a similarity score function with a corresponding threshold. First we discuss the TriviaQA dataset, which contains short answers. For such cases, typically the F1 score is used in the literature (Joshi et al., 2017; Devlin et al., 2019). However, to better accommodate the case that the response may be long and the answer (label) is very short, which significantly reduces the F1 score, we rather consider recall as the similarity score in the experiments. To select the threshold, we uniformly sampled 100 question-answer pairs that were not used for calibration or testing, and manually inspected the similarity of the generated response and the true answer. With respective thresholds of 0.5 and 0.25, the recall and F1 scores make no mistakes, hence, in the experiments we used recall with threshold 0.5 as the match function for this dataset. According to (6), this implies that any measurement of the error rate in testing with this match function would result in at most 1 percentage-point lower error in expectation than the ground truth.

Selecting an appropriate similarity function for the Temporal Sequences dataset is much harder because it has a large proportion of long answer, which makes the F1 and recall scores much less useful: with the same thresholds as above, on a random sample of 100 question-answer pairs, the F1 score resulted in 45 mistakes while and recall score ended up with 13, after manually checking the validity of the corresponding responses generated by the language model. Therefore, we decided to prompt the LLM to compute the similarity of the response and the true answer, using the similarity-seeking prompt presented in Appendix B. Then we computed the smallest threshold (which was $\hat{\beta} = 7$ in this case) so that the number of errors in the 100 datapoints was 4 (as verified by manual inspection); according to (6), this guarantees that the (expected) error rate as measured by the resulting match function (i.e., using the $\hat{\beta}$ -thresholded LLM self-prompting score) is at most 5 percentage-point lower than the true error rate. Therefore, for the Temporal Sequences dataset we used the LLM self-prompting similarity score with threshold $\hat{\beta} = 7$ to compute the match function.

(Note that the same method resulted in 2 mistakes for the TriviaQA dataset.)

Calibration methods. For calibrated methods with theoretical guarantees, we consider the CRC method (defined in (3), and referred to as ‘Bound in expectation’), and three variants of the RCPS procedure, as described in Section 4, with UCB given by $\hat{R}_{\text{ub-bern}}$ (referred to as ‘Emp. Bernstein’), $\hat{R}_{\text{ub-hb}}$ (referred to as ‘Hoeffding-Bentkus’), $\hat{R}_{\text{ub-kl}}$ (referred to as ‘Bernoulli KL’).

For a given risk tolerance α , a simple baseline abstention policy chooses the smallest parameter λ that satisfies $L_n(\lambda) \leq \alpha$. We do not however have a theoretical guarantee on the risk of this baseline, and as we will show, it might violate the risk condition with small calibration datasets. This method is referred to as ‘Baseline’ in the experiments.

Note that we do not include the high-probability amplification of CRC discussed in Proposition 3.1 in our experiments. The risk guarantee provided by this bound is $\mathbb{P}(R(\hat{\lambda}^*) \leq e\alpha) \geq 1 - \delta$, i.e. the bound is inflated by e compared to RCPS baselines. So, to properly compare it to other baselines we need to replace α by α/e , which makes the guarantee quite conservative and results in a very high abstention rate.

Score functions. We consider calibration of three different scores using the above methods. The first two scores are the *greedy* variants of the score functions proposed and described in Section 2.1: In both cases, we take the greedy (zero-temperature) response as the reference response and sample additional $k - 1 = 10$ extra responses at temperature 0.9. Then we either (i) prompt the LLM for the similarity of the reference response and each of the extra responses, and obtain the number of matches between the reference response and the extra responses — this is referred to as *match count (m.c.)* in the results; (ii) prompt the LLM for the number of matches between the reference response and the extra responses at once, and calculate the expected number of matches using the log-probabilities assigned by the LLM to the responses “1”, “2”, ... — this is referred to as *expected match count (e.m.c.)* in the results.

The third scoring method is the simple baseline of the log-probability of the zero-temperature response. Another popular score in the literature is the normalized log-probability; however, we only report results with the log-probability score, as in our experiments it always performed at least as good as its normalized version. This baseline is referred to as *log-probabilities (l.p.)* in the results.

The prompts used in calculating the score functions and some data samples are described in Appendix B.

7.1 Results on the Temporal Sequences dataset

We experimented with 4000 question-answer pairs. The experiments were performed with two risk tolerance levels, $\alpha = 0.05$ and $\alpha = 0.1$, and we used $\delta = 0.05$ failure probability for confidence intervals.

The results of the experiments are reported in Figure 1, which shows the average test losses vs. abstention rates on the test sample for calibration datasets of various sizes (the exact numerical results are reported in Tables 1 to 4, 9 and 10 in Appendix C). As expected, we can observe an inherent trade-off between the two metrics: in particular, a larger abstention rate leads to a smaller test error; however, some methods and baselines exhibit better trade-offs. For instance, by looking at Figure 1 we can observe that for a sufficiently large calibration sample, it is evident that log-probability scoring performs considerably *worse* regardless of which conformal prediction method (CRC/RCPS, confidence bound) is used. At the same time, the proposed match count (m.c.) and expected match count (e.m.c.) proposed perform much better, and the difference between the CRC and RCPS methods is minimal.

We also observe that the Empirical Bernstein calibration method is significantly worse than the others; this is expected since here we estimate Bernoulli random variables, and the other two bounds used in the RCPS methods are specialized for this case, unlike the Empirical Bernstein bound, which – unlike the other two – would be applicable for non-binary loss functions, as well. We can also observe that the uncalibrated Baseline methods violate the risk conditions for smaller calibration datasets a bit more than other methods.

7.2 Results on the TriviaQA dataset

We experimented on the TriviaQA dataset in a similar fashion. In particular, we used 1000 randomly selected question-answer pairs, performed experiments with two risk tolerance levels, $\alpha = 0.1$ and $\alpha = 0.2$, and used $\delta = 0.05$ failure probability for the confidence intervals.

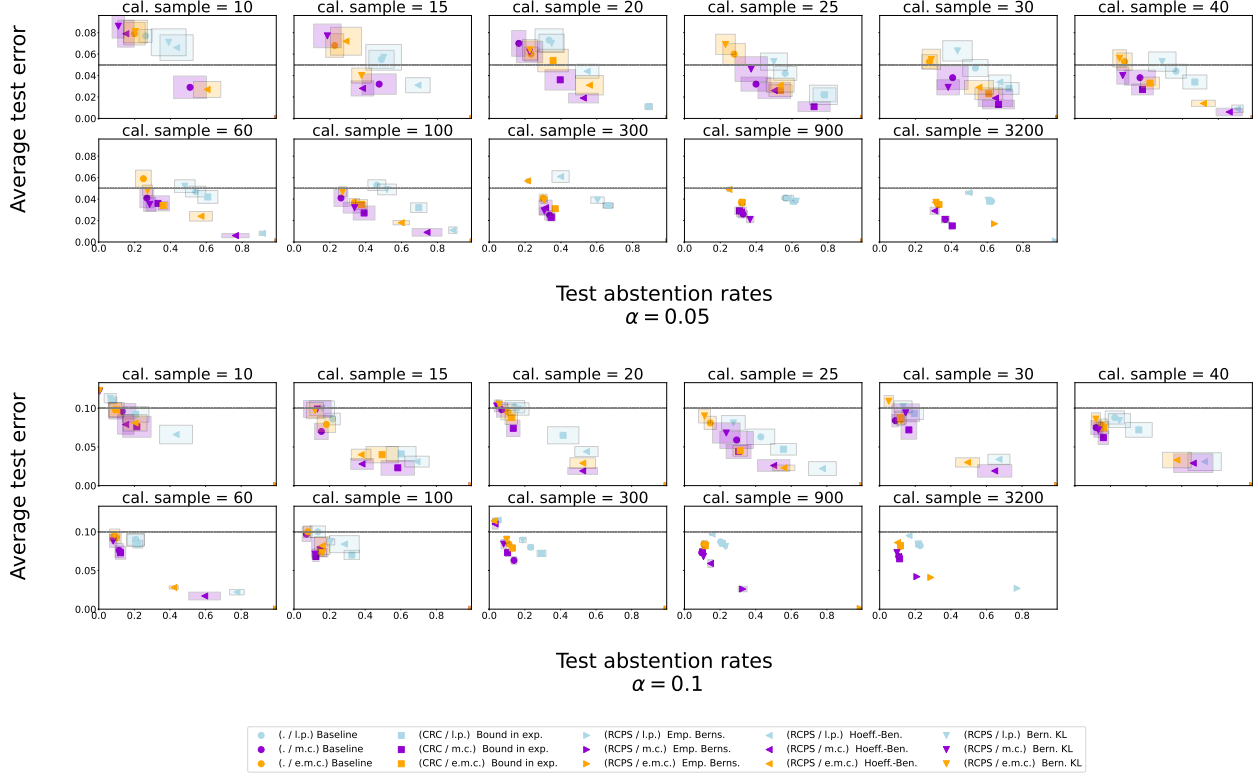


Figure 1: Abstention rates vs. average test losses on the Temporal Sequences dataset with $\alpha = 0.05$ (top) and $\alpha = 0.1$ (bottom) for score functions match count (m.c.), expected match count (e.m.c), and the log-probability (l.p.), and for various calibration methods (. denotes the baseline with no calibration). Box widths and heights represent 90% confidence intervals with Gaussian approximation over abstention rates and average test errors, respectively. The dashed horizontal line represents the target risk bound α .

Similarly to our other experiment, Figure 2 shows the trade-off between the abstention rate and test error. As a result of the fact that the LLM tends to generate shorter responses on the queries in this dataset (and the true responses are also short), log-probability scoring is competitive with our proposed scoring methods. In fact, they seem to perform quite similarly in all experiments (with the log-probability scores being somewhat better for $\alpha = 0.1$ and worse for $\alpha = 0.2$). As before, we observe that there is a negligible difference between the CRC and RCPS methods, and that Baseline sometimes violates the risk condition with smaller calibration datasets.

More details (with the exact numerical results) are presented in Tables 5 to 8, 11 and 12 in Appendix C.

Comparing the experiments for the two datasets, we can conclude that our proposed calibrated abstention methods based on match counts (or expected match counts) are preferable to the variant based on log-probability, as they perform well for both short and long answers, while the log-probability score is significantly worse for questions with long answers.

8 Conclusions and future directions

We proposed a conformal calibration and similarity scoring procedure which enables LLMs to abstain in a principled way. In particular, one of our main contributions is a novel procedure to generate match scores to count the number of similar responses to a query. When combined with conformal calibration, this scoring

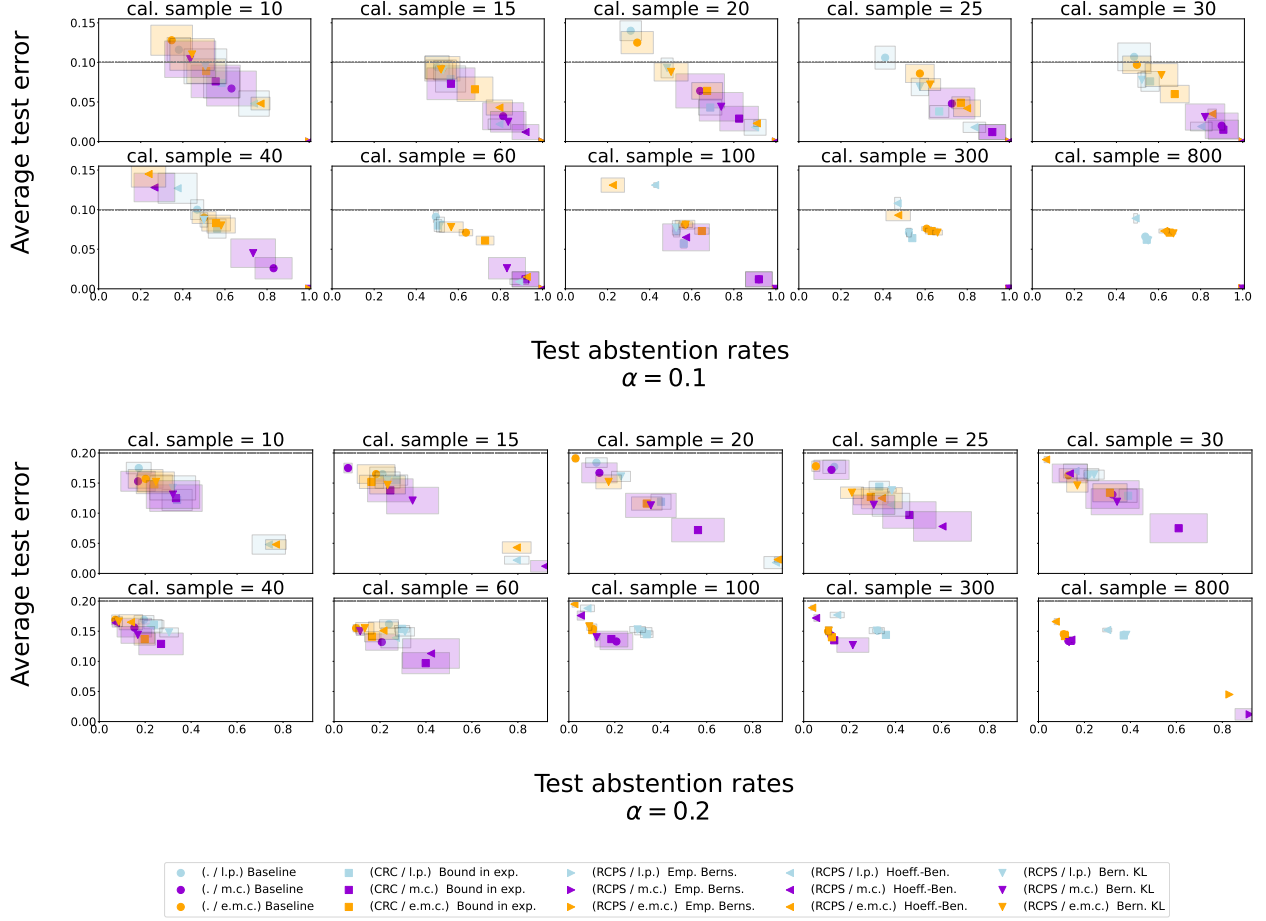


Figure 2: Abstention rates vs. average test losses on the TriviaQA dataset with $\alpha = 0.1$ (top) and $\alpha = 0.2$ (bottom) for score functions match count (m.c.), expected match count (e.m.c), and the log-probability (l.p.), and for various calibration methods (. denotes the baseline with no calibration). Box widths and heights represent 90% confidence intervals with Gaussian approximation over abstention rates and average test errors, respectively. The dashed horizontal line represents the target risk bound α .

procedure achieves a good trade-off between abstention rate and test performance. Importantly, in experiments over two question-answering datasets, our proposed procedure surpasses the simple baseline scoring procedure of using log-probabilities of the predictor (once more suggesting that LLMs are not well-calibrated). Finally, we also presented a method to calibrate the match function (based on similarity measures) which is used in automatically evaluating the performance of the LLM at test time, which comes with theoretical guarantees on its accuracy and requires only a small labelled calibration set to tune the threshold.

References

- Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.
- Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations (ICLR)*, 2024.

- Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Tuning bandit algorithms in stochastic environments. In *Algorithmic Learning Theory (ALT)*, pages 150–165. Springer, 2007.
- Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it is lying. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.
- Jeremy R. Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, 2019.
- Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 2010.
- Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- Gemini Team, Google. Gemini: A family of highly capable multimodal models. https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf, 2023. [Online; accessed 01-February-2024].
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Transactions of the Association for Computational Linguistics (ACL)*, pages 1601–1611, 2017.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, and et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Transactions of the Association for Computational Linguistics (ACL)*, 2020.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.
- Kwang-Sung and Francesco Orabona. Parameter-free online convex optimization with sub-exponential noise. In *Conference on Computational Learning Theory (COLT)*, 2019.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- Zhen Lin, Lucas Glass, M Brandon Westover, Cao Xiao, and Jimeng Sun. Scrib: set-classifier with class-specific risk bounds for blackbox models. In *Conference on Artificial Intelligence (AAAI)*, volume 36, pages 7497–7505, 2022.
- Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

- Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations (ICLR)*, 2020.
- Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- Andreas Maurer. A note on the PAC Bayesian theorem. *arXiv preprint cs/0411099*, 2004.
- Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Conference on Computational Learning Theory (COLT)*, 2009.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Transactions of the Association for Computational Linguistics (ACL)*, pages 1906–1919, 2020.
- Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents’ overconfidence through linguistic calibration. In *Transactions of the Association for Computational Linguistics (ACL)*, 2022.
- Francesco Orabona and Kwang-Sung Jun. Tight concentrations and confidence sequences from the regret of universal portfolio. *IEEE Transactions on Information Theory*, 2023.
- Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.
- Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. [abs/2306.10193](https://arxiv.org/abs/2306.10193), 2023.
- Stephan Rabanser, Anvith Thudi, Kimia Hamidieh, Adam Dziedzic, and Nicolas Papernot. Selective classification via neural network training dynamics. *arXiv preprint arXiv:2205.13532*, 2022.
- Shauli Ravfogel, Yoav Goldberg, and Jacob Goldberger. Conformal nucleus sampling. In *Transactions of the Association for Computational Linguistics (ACL)*, pages 27–34. Association for Computational Linguistics, 2023.
- Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning*, pages 661–682. PMLR, 2023.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research (TMLR)*, 2023.
- Robert J Tibshirani and Bradley Efron. An introduction to the bootstrap. *Monographs on statistics and applied probability*, 57(1), 1993.
- Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2022.

A CRC in expectation

Lemma A.1. Assume that $\lambda \mapsto \ell(z, \lambda)$ is non-increasing and upper-bounded by B for any example $z \in \mathcal{Z}$. Then, for

$$\hat{\lambda} = \inf \left\{ \lambda \in \Lambda : L_n(\lambda) + \frac{B}{n} \leq \alpha \right\} \quad (\alpha > 0)$$

we have $(1 + \frac{1}{n})\mathbb{E}R(\hat{\lambda}) \leq \alpha$ assuming that D_n is drawn i.i.d.

Proof. For some error $\alpha > 0$ and a free parameter $\epsilon > 0$ (to be tuned later) consider solution

$$\hat{\lambda} = \inf \{ \lambda \in \Lambda : L(D_n; \lambda) + \epsilon \leq \alpha \} .$$

Let $\tilde{Z}_i$ be an independent copy of Z_i and suppose that $\hat{\lambda}^{+i}$ is a solution obtained by adding loss $\ell(\tilde{Z}_i, \lambda)/n$ to the objective, namely

$$\hat{\lambda}^{+i} = \inf \left\{ \lambda \in \Lambda : L(D_n; \lambda) + \frac{\ell(\tilde{Z}_i, \lambda)}{n} \leq \alpha \right\} . \quad (7)$$

Choosing $\epsilon = B/n$, we observe that $\hat{\lambda}^{+i} \leq \hat{\lambda}$ for any i (the feasible set of $\hat{\lambda}^{+i}$ is no smaller than that of $\hat{\lambda}$). Hence by the non-increasing property of the loss,

$$\ell(\tilde{Z}_i, \hat{\lambda}^{+i}) \geq \ell(\tilde{Z}_i, \hat{\lambda})$$

while summing over losses w.r.t $(\tilde{Z}_1, \dots, \tilde{Z}_n, \tilde{Z}_i)$ and taking expectation gives

$$\frac{1}{n} \sum_{j=1}^n \mathbb{E}[\ell(\tilde{Z}_j, \hat{\lambda}^{+i})] + \frac{\mathbb{E}[\ell(\tilde{Z}_i, \hat{\lambda}^{+i})]}{n} \geq \frac{1}{n} \sum_{j=1}^n \mathbb{E}[\ell(\tilde{Z}_j, \hat{\lambda})] + \frac{\mathbb{E}[\ell(\tilde{Z}_i, \hat{\lambda})]}{n} = (1 + \frac{1}{n})\mathbb{E}L_n(\hat{\lambda}) .$$

On the other hand, by identity (or exchangeability) and using the fact that $\hat{\lambda}^{+i}$ is a solution to Eq. (7)

$$\frac{1}{n} \sum_{j=1}^n \mathbb{E}[\ell(Z_j, \hat{\lambda}^{+i})] + \frac{\mathbb{E}[\ell(\tilde{Z}_i, \hat{\lambda}^{+i})]}{n} \leq \alpha \quad \Leftrightarrow \quad \frac{1}{n} \sum_{j=1}^n \mathbb{E}[\ell(\tilde{Z}_j, \hat{\lambda}^{+i})] + \frac{\mathbb{E}[\ell(\tilde{Z}_i, \hat{\lambda}^{+i})]}{n} \leq \alpha .$$

□

B Details of the experimental setup

LLM prompting. We use the following prompt to query the model.

Answer the following question: Q: {query}
A:

We use the following prompt to query the model for the similarity of two responses given a question (used in computing the match count score):

You are given a question (Q), and two candidate answers (C and D). Given Q, measure the similarity of C and D on a scale of 0-10. Answer in a number between 0 and 10.

Q: {question}

C: {response1}
D: {response2}

Given Q, similarity of C and D to each other is [0-10]:

We also consider a more efficient implementation with a single prompt that queries similarity of one response with $K = 10$ other responses (used in computing the expected match count score):

You are given a question (Q), a candidate answer (C), and a list of alternative answers (A1-A10). How many of the alternative answers are semantically the same as C, if any? Generic answers do not count. Answer in a number between 0 and 10.

Q: {question}

C: {response}

A1: {response1}
A2: {response2}
A3: {response3}
A4: {response4}
A5: {response5}
A6: {response6}
A7: {response7}
A8: {response8}
A9: {response9}
A10: {response10}

The number of supportive answers for C is [0-10]:

Data. We perform experiments on a subset of a TriviaQA dataset, and on the full Temporal Sequences dataset (from BIG-bench). Below is a sample from Temporal Sequences dataset when the model generates a long answer.

Q: Today, Jennifer went to the library. Between what times could they have gone? We know that: Jennifer woke up at 5am. William saw Jennifer working at the office from 5am to 8am. David saw Jennifer taking photos near the Leaning Tower of Pisa from 8am to 10am. Jason saw Jennifer driving to the water park from 10am to 1pm. Sarah saw Jennifer stretching at a yoga studio from 4pm to 7pm. The library was closed after 7pm. Between what times could Jennifer have gone to the library?

A: 1pm to 4pm

Temp=0 response: Jennifer could have gone to the library between 1pm and 4pm. Jennifer woke up at 5am and was seen working at the office from 5am to 8am by William. From 8am to 10am, David saw her taking photos near the Leaning Tower of Pisa. From 10am to 1pm, Jason saw her driving to the water park. This means that Jennifer was not at the library during these times. Sarah saw Jennifer stretching at a yoga studio from 4pm to 7pm. The library was closed after 7pm. This means that Jennifer could not have gone to the library after 4pm. Therefore, the only time Jennifer could have gone to the library is between 1pm and 4pm.

Table 4: Dataset Temporal Sequences (temp. = 0 response): AVERAGE TEST ABSTENTION RATES. $\alpha = 0.05$

baseline / sample size	10	15	20	25	30	40	60	100	300	900	3200
(. / e.m.c.) Baseline	0.199 $\pm$ 0.120	0.227 $\pm$ 0.100	0.233 $\pm$ 0.076	0.277 $\pm$ 0.125	0.278 $\pm$ 0.085	0.277 $\pm$ 0.085	0.249 $\pm$ 0.088	0.337 $\pm$ 0.056	0.301 $\pm$ 0.043	0.318 $\pm$ 0.019	0.321 $\pm$ 0.012
(CRC / e.m.c.) Bound in expectation	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.356 $\pm$ 0.170	0.536 $\pm$ 0.129	0.609 $\pm$ 0.155	0.419 $\pm$ 0.106	0.359 $\pm$ 0.053	0.376 $\pm$ 0.067	0.368 $\pm$ 0.030	0.322 $\pm$ 0.020	0.331 $\pm$ 0.011
(. / m.c.) Baseline	0.509 $\pm$ 0.183	0.475 $\pm$ 0.190	0.165 $\pm$ 0.079	0.399 $\pm$ 0.167	0.407 $\pm$ 0.172	0.363 $\pm$ 0.171	0.268 $\pm$ 0.077	0.262 $\pm$ 0.077	0.337 $\pm$ 0.055	0.327 $\pm$ 0.050	0.366 $\pm$ 0.041
(CRC / m.c.) Bound in expectation	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.396 $\pm$ 0.170	0.723 $\pm$ 0.179	0.663 $\pm$ 0.179	0.377 $\pm$ 0.121	0.330 $\pm$ 0.130	0.391 $\pm$ 0.121	0.346 $\pm$ 0.047	0.307 $\pm$ 0.051	0.405 $\pm$ 0.000
(. / l.p.) Baseline	0.261 $\pm$ 0.073	0.488 $\pm$ 0.141	0.334 $\pm$ 0.157	0.562 $\pm$ 0.129	0.534 $\pm$ 0.142	0.564 $\pm$ 0.116	0.540 $\pm$ 0.113	0.462 $\pm$ 0.100	0.662 $\pm$ 0.056	0.565 $\pm$ 0.043	0.624 $\pm$ 0.022
(CRC / l.p.) Bound in expectation	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.890 $\pm$ 0.043	0.781 $\pm$ 0.127	0.721 $\pm$ 0.113	0.669 $\pm$ 0.134	0.608 $\pm$ 0.112	0.696 $\pm$ 0.091	0.664 $\pm$ 0.053	0.609 $\pm$ 0.041	0.614 $\pm$ 0.020
(RCPS / e.m.c.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.640 $\pm$ 0.028
(RCPS / e.m.c.) Hoeffding-Bentkus	0.604 $\pm$ 0.153	0.290 $\pm$ 0.173	0.559 $\pm$ 0.155	0.534 $\pm$ 0.155	0.554 $\pm$ 0.162	0.717 $\pm$ 0.132	0.570 $\pm$ 0.127	0.598 $\pm$ 0.091	0.214 $\pm$ 0.014	0.248 $\pm$ 0.020	0.311 $\pm$ 0.011
(RCPS / e.m.c.) Bernoulli KL	0.204 $\pm$ 0.134	0.377 $\pm$ 0.110	0.227 $\pm$ 0.084	0.229 $\pm$ 0.109	0.288 $\pm$ 0.102	0.249 $\pm$ 0.076	0.272 $\pm$ 0.055	0.271 $\pm$ 0.046	0.306 $\pm$ 0.034	0.323 $\pm$ 0.024	0.314 $\pm$ 0.014
(RCPS / m.c.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.405 $\pm$ 0.000
(RCPS / m.c.) Hoeffding-Bentkus	0.151 $\pm$ 0.092	0.382 $\pm$ 0.126	0.525 $\pm$ 0.167	0.501 $\pm$ 0.181	0.646 $\pm$ 0.192	0.861 $\pm$ 0.146	0.762 $\pm$ 0.152	0.742 $\pm$ 0.166	0.314 $\pm$ 0.073	0.318 $\pm$ 0.057	0.307 $\pm$ 0.051
(RCPS / m.c.) Bernoulli KL	0.109 $\pm$ 0.069	0.186 $\pm$ 0.153	0.225 $\pm$ 0.146	0.372 $\pm$ 0.180	0.382 $\pm$ 0.126	0.267 $\pm$ 0.067	0.284 $\pm$ 0.067	0.338 $\pm$ 0.126	0.307 $\pm$ 0.051	0.366 $\pm$ 0.041	0.366 $\pm$ 0.041
(RCPS / l.p.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.980 $\pm$ 0.005
(RCPS / l.p.) Hoeffding-Bentkus	0.434 $\pm$ 0.185	0.691 $\pm$ 0.137	0.546 $\pm$ 0.131	0.775 $\pm$ 0.156	0.670 $\pm$ 0.124	0.907 $\pm$ 0.064	0.912 $\pm$ 0.035	0.886 $\pm$ 0.047	0.398 $\pm$ 0.096	0.579 $\pm$ 0.058	0.498 $\pm$ 0.038
(RCPS / l.p.) Bernoulli KL	0.390 $\pm$ 0.194	0.500 $\pm$ 0.199	0.348 $\pm$ 0.140	0.499 $\pm$ 0.150	0.434 $\pm$ 0.165	0.487 $\pm$ 0.154	0.480 $\pm$ 0.115	0.519 $\pm$ 0.105	0.604 $\pm$ 0.075	0.624 $\pm$ 0.022	0.605 $\pm$ 0.015

Table 5: Dataset TriviaQA (temp. = 0 response): AVERAGE TEST LOSSES. $\alpha = 0.1$

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	0.128 $\pm$ 0.038	0.092 $\pm$ 0.031	0.125 $\pm$ 0.028	0.086 $\pm$ 0.023	0.097 $\pm$ 0.022	0.090 $\pm$ 0.015	0.071 $\pm$ 0.008	0.081 $\pm$ 0.011	0.076 $\pm$ 0.005	0.072 $\pm$ 0.002
(CRC / e.m.c.) Bound in expectation	0.089 $\pm$ 0.031	0.066 $\pm$ 0.031	0.064 $\pm$ 0.021	0.049 $\pm$ 0.015	0.060 $\pm$ 0.022	0.083 $\pm$ 0.020	0.061 $\pm$ 0.011	0.073 $\pm$ 0.008	0.073 $\pm$ 0.007	0.071 $\pm$ 0.003
(. / m.c.) Baseline	0.067 $\pm$ 0.044	0.032 $\pm$ 0.034	0.064 $\pm$ 0.043	0.048 $\pm$ 0.039	0.020 $\pm$ 0.032	0.026 $\pm$ 0.027	0.012 $\pm$ 0.019	0.012 $\pm$ 0.019	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(CRC / m.c.) Bound in expectation	0.076 $\pm$ 0.041	0.073 $\pm$ 0.040	0.029 $\pm$ 0.031	0.012 $\pm$ 0.019	0.015 $\pm$ 0.024	0.000 $\pm$ 0.000	0.012 $\pm$ 0.019	0.012 $\pm$ 0.019	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(. / l.p.) Baseline	0.116 $\pm$ 0.031	0.094 $\pm$ 0.029	0.140 $\pm$ 0.025	0.106 $\pm$ 0.029	0.107 $\pm$ 0.035	0.100 $\pm$ 0.025	0.091 $\pm$ 0.014	0.073 $\pm$ 0.015	0.071 $\pm$ 0.010	0.066 $\pm$ 0.002
(CRC / l.p.) Bound in expectation	0.074 $\pm$ 0.038	0.079 $\pm$ 0.033	0.043 $\pm$ 0.019	0.038 $\pm$ 0.015	0.076 $\pm$ 0.025	0.076 $\pm$ 0.024	0.080 $\pm$ 0.015	0.057 $\pm$ 0.011	0.064 $\pm$ 0.007	0.062 $\pm$ 0.005
(RCPS / e.m.c.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.001 $\pm$ 0.001
(RCPS / e.m.c.) Hoeffding-Bentkus	0.048 $\pm$ 0.016	0.043 $\pm$ 0.019	0.023 $\pm$ 0.008	0.042 $\pm$ 0.021	0.035 $\pm$ 0.012	0.145 $\pm$ 0.033	0.015 $\pm$ 0.005	0.131 $\pm$ 0.018	0.093 $\pm$ 0.014	0.073 $\pm$ 0.005
(RCPS / e.m.c.) Bernoulli KL	0.110 $\pm$ 0.040	0.091 $\pm$ 0.024	0.088 $\pm$ 0.023	0.072 $\pm$ 0.015	0.084 $\pm$ 0.029	0.080 $\pm$ 0.019	0.078 $\pm$ 0.013	0.081 $\pm$ 0.007	0.071 $\pm$ 0.006	0.070 $\pm$ 0.002
(RCPS / m.c.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(RCPS / m.c.) Hoeffding-Bentkus	0.000 $\pm$ 0.000	0.012 $\pm$ 0.019	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.128 $\pm$ 0.036	0.000 $\pm$ 0.000	0.065 $\pm$ 0.034	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(RCPS / m.c.) Bernoulli KL	0.104 $\pm$ 0.046	0.025 $\pm$ 0.026	0.044 $\pm$ 0.036	0.012 $\pm$ 0.019	0.031 $\pm$ 0.033	0.045 $\pm$ 0.036	0.026 $\pm$ 0.027	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(RCPS / l.p.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(RCPS / l.p.) Hoeffding-Bentkus	0.048 $\pm$ 0.032	0.022 $\pm$ 0.013	0.018 $\pm$ 0.020	0.018 $\pm$ 0.013	0.019 $\pm$ 0.011	0.127 $\pm$ 0.038	0.009 $\pm$ 0.007	0.131 $\pm$ 0.006	0.108 $\pm$ 0.015	0.089 $\pm$ 0.012
(RCPS / l.p.) Bernoulli KL	0.095 $\pm$ 0.044	0.090 $\pm$ 0.024	0.093 $\pm$ 0.025	0.069 $\pm$ 0.022	0.078 $\pm$ 0.019	0.087 $\pm$ 0.018	0.081 $\pm$ 0.019	0.077 $\pm$ 0.018	0.071 $\pm$ 0.011	0.061 $\pm$ 0.003

Table 6: Dataset TriviaQA (temp. = 0 response): AVERAGE TEST ABSTENTION RATES. $\alpha = 0.1$

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	0.347 $\pm$ 0.198	0.524 $\pm$ 0.167	0.341 $\pm$ 0.153	0.575 $\pm$ 0.136	0.497 $\pm$ 0.135	0.501 $\pm$ 0.114	0.636 $\pm$ 0.064	0.568 $\pm$ 0.093	0.607 $\pm$ 0.047	0.643 $\pm$ 0.025
(CRC / e.m.c.) Bound in expectation	0.511 $\pm$ 0.185	0.679 $\pm$ 0.170	0.673 $\pm$ 0.146	0.770 $\pm$ 0.086	0.677 $\pm$ 0.137	0.558 $\pm$ 0.136	0.727 $\pm$ 0.084	0.648 $\pm$ 0.068	0.631 $\pm$ 0.060	0.653 $\pm$ 0.029
(. / m.c.) Baseline	0.631 $\pm$ 0.236	0.813 $\pm$ 0.195	0.640 $\pm$ 0.231	0.727 $\pm$ 0.218	0.900 $\pm$ 0.156	0.831 $\pm$ 0.176	0.919 $\pm$ 0.127	0.919 $\pm$ 0.127	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
(CRC / m.c.) Bound in expectation	0.556 $\pm$ 0.233	0.565 $\pm$ 0.228	0.825 $\pm$ 0.183	0.919 $\pm$ 0.127	0.908 $\pm$ 0.143	1.000 $\pm$ 0.000	0.919 $\pm$ 0.127	0.919 $\pm$ 0.127	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
(. / l.p.) Baseline	0.381 $\pm$ 0.111	0.499 $\pm$ 0.114	0.310 $\pm$ 0.102	0.410 $\pm$ 0.096	0.484 $\pm$ 0.155	0.468 $\pm$ 0.062	0.493 $\pm$ 0.023	0.528 $\pm$ 0.029	0.523 $\pm$ 0.020	0.537 $\pm$ 0.006
(CRC / l.p.) Bound in expectation	0.584 $\pm$ 0.171	0.563 $\pm$ 0.143	0.688 $\pm$ 0.110	0.667 $\pm$ 0.091	0.560 $\pm$ 0.087	0.563 $\pm$ 0.084	0.503 $\pm$ 0.035	0.562 $\pm$ 0.027	0.539 $\pm$ 0.016	0.544 $\pm$ 0.012
(RCPS / e.m.c.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.998 $\pm$ 0.002
(RCPS / e.m.c.) Hoeffding-Bentkus	0.769 $\pm$ 0.090	0.795 $\pm$ 0.125	0.909 $\pm$ 0.029	0.799 $\pm$ 0.128	0.855 $\pm$ 0.041	0.237 $\pm$ 0.163	0.925 $\pm$ 0.020	0.225 $\pm$ 0.111	0.472 $\pm$ 0.116	0.627 $\pm$ 0.050
(RCPS / e.m.c.) Bernoulli KL	0.444 $\pm$ 0.211	0.518 $\pm$ 0.145	0.502 $\pm$ 0.161	0.624 $\pm$ 0.099	0.614 $\pm$ 0.155	0.582 $\pm$ 0.134	0.566 $\pm$ 0.114	0.571 $\pm$ 0.066	0.659 $\pm$ 0.046	0.671 $\pm$ 0.023
(RCPS / m.c.) Emp. Bernstein										

Table 7: Dataset TriviaQA (temp. = 0 response): AVERAGE TEST LOSSES. $\alpha = 0.2$ (m.c. = match counts, e.m.c. = expected match counts, l.p. = log-probabilities)

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	0.157 $\pm$ 0.031	0.165 $\pm$ 0.031	0.191 $\pm$ 0.011	0.178 $\pm$ 0.017	0.163 $\pm$ 0.022	0.169 $\pm$ 0.016	0.155 $\pm$ 0.012	0.154 $\pm$ 0.016	0.150 $\pm$ 0.012	0.145 $\pm$ 0.004
(CRC / e.m.c.) Bound in expectation	0.148 $\pm$ 0.031	0.152 $\pm$ 0.022	0.116 $\pm$ 0.022	0.127 $\pm$ 0.026	0.134 $\pm$ 0.030	0.137 $\pm$ 0.020	0.141 $\pm$ 0.017	0.152 $\pm$ 0.011	0.140 $\pm$ 0.004	0.142 $\pm$ 0.005
(. / m.c.) Baseline	0.153 $\pm$ 0.033	0.175 $\pm$ 0.016	0.167 $\pm$ 0.033	0.172 $\pm$ 0.032	0.131 $\pm$ 0.046	0.156 $\pm$ 0.031	0.132 $\pm$ 0.027	0.133 $\pm$ 0.027	0.143 $\pm$ 0.010	0.134 $\pm$ 0.004
(CRC / m.c.) Bound in expectation	0.125 $\pm$ 0.044	0.138 $\pm$ 0.038	0.072 $\pm$ 0.039	0.097 $\pm$ 0.044	0.075 $\pm$ 0.048	0.129 $\pm$ 0.036	0.097 $\pm$ 0.035	0.137 $\pm$ 0.026	0.135 $\pm$ 0.005	0.134 $\pm$ 0.006
(. / l.p.) Baseline	0.175 $\pm$ 0.019	0.165 $\pm$ 0.023	0.184 $\pm$ 0.016	0.177 $\pm$ 0.022	0.164 $\pm$ 0.020	0.169 $\pm$ 0.014	0.162 $\pm$ 0.014	0.150 $\pm$ 0.012	0.151 $\pm$ 0.006	0.144 $\pm$ 0.000
(CRC / l.p.) Bound in expectation	0.122 $\pm$ 0.035	0.155 $\pm$ 0.017	0.119 $\pm$ 0.025	0.144 $\pm$ 0.018	0.129 $\pm$ 0.022	0.161 $\pm$ 0.014	0.154 $\pm$ 0.011	0.145 $\pm$ 0.011	0.144 $\pm$ 0.003	0.143 $\pm$ 0.003
(RCPS / e.m.c.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.045 $\pm$ 0.003
(RCPS / e.m.c.) Hoeffing-Bentkus	0.048 $\pm$ 0.016	0.043 $\pm$ 0.019	0.023 $\pm$ 0.008	0.125 $\pm$ 0.035	0.189 $\pm$ 0.010	0.165 $\pm$ 0.025	0.151 $\pm$ 0.029	0.195 $\pm$ 0.010	0.189 $\pm$ 0.008	0.166 $\pm$ 0.010
(RCPS / e.m.c.) Bernoulli KL	0.152 $\pm$ 0.033	0.147 $\pm$ 0.030	0.152 $\pm$ 0.023	0.134 $\pm$ 0.019	0.146 $\pm$ 0.022	0.166 $\pm$ 0.016	0.155 $\pm$ 0.019	0.158 $\pm$ 0.013	0.151 $\pm$ 0.014	0.144 $\pm$ 0.004
(RCPS / m.c.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.012 $\pm$ 0.019
(RCPS / m.c.) Hoeffing-Bentkus	0.000 $\pm$ 0.000	0.012 $\pm$ 0.019	0.000 $\pm$ 0.000	0.078 $\pm$ 0.049	0.166 $\pm$ 0.031	0.164 $\pm$ 0.014	0.113 $\pm$ 0.049	0.176 $\pm$ 0.015	0.172 $\pm$ 0.011	0.136 $\pm$ 0.009
(RCPS / m.c.) Bernoulli KL	0.131 $\pm$ 0.046	0.121 $\pm$ 0.044	0.113 $\pm$ 0.040	0.114 $\pm$ 0.032	0.119 $\pm$ 0.043	0.144 $\pm$ 0.028	0.150 $\pm$ 0.014	0.140 $\pm$ 0.009	0.127 $\pm$ 0.024	0.132 $\pm$ 0.002
(RCPS / l.p.) Emp. Bernstein	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
(RCPS / l.p.) Hoeffing-Bentkus	0.048 $\pm$ 0.032	0.022 $\pm$ 0.013	0.018 $\pm$ 0.020	0.114 $\pm$ 0.038	0.170 $\pm$ 0.023	0.159 $\pm$ 0.021	0.139 $\pm$ 0.033	0.188 $\pm$ 0.011	0.177 $\pm$ 0.009	0.152 $\pm$ 0.006
(RCPS / l.p.) Bernoulli KL	0.142 $\pm$ 0.040	0.155 $\pm$ 0.027	0.161 $\pm$ 0.017	0.138 $\pm$ 0.006	0.164 $\pm$ 0.015	0.148 $\pm$ 0.015	0.149 $\pm$ 0.014	0.153 $\pm$ 0.010	0.151 $\pm$ 0.010	0.144 $\pm$ 0.000

Table 8: Dataset TriviaQA (temp. = 0 response): AVERAGE TEST ABSTENTION RATES. $\alpha = 0.2$ (m.c. = match counts, e.m.c. = expected match counts, l.p. = log-probabilities)

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	0.202 $\pm$ 0.151	0.183 $\pm$ 0.164	0.029 $\pm$ 0.021	0.053 $\pm$ 0.035	0.130 $\pm$ 0.080	0.070 $\pm$ 0.031	0.096 $\pm$ 0.026	0.105 $\pm$ 0.033	0.106 $\pm$ 0.020	0.110 $\pm$ 0.010
(CRC / e.m.c.) Bound in expectation	0.240 $\pm$ 0.157	0.163 $\pm$ 0.105	0.339 $\pm$ 0.133	0.292 $\pm$ 0.150	0.311 $\pm$ 0.149	0.198 $\pm$ 0.092	0.165 $\pm$ 0.065	0.102 $\pm$ 0.024	0.123 $\pm$ 0.014	0.115 $\pm$ 0.012
(. / m.c.) Baseline	0.169 $\pm$ 0.150	0.061 $\pm$ 0.036	0.133 $\pm$ 0.154	0.121 $\pm$ 0.153	0.322 $\pm$ 0.232	0.153 $\pm$ 0.150	0.208 $\pm$ 0.141	0.207 $\pm$ 0.142	0.120 $\pm$ 0.026	0.130 $\pm$ 0.016
(CRC / m.c.) Bound in expectation	0.334 $\pm$ 0.228	0.245 $\pm$ 0.198	0.562 $\pm$ 0.229	0.460 $\pm$ 0.232	0.610 $\pm$ 0.249	0.269 $\pm$ 0.193	0.399 $\pm$ 0.207	0.185 $\pm$ 0.143	0.133 $\pm$ 0.020	0.140 $\pm$ 0.021
(. / l.p.) Baseline	0.172 $\pm$ 0.108	0.211 $\pm$ 0.111	0.120 $\pm$ 0.094	0.133 $\pm$ 0.099	0.214 $\pm$ 0.097	0.195 $\pm$ 0.086	0.239 $\pm$ 0.082	0.324 $\pm$ 0.075	0.322 $\pm$ 0.044	0.372 $\pm$ 0.008
(CRC / l.p.) Bound in expectation	0.339 $\pm$ 0.131	0.272 $\pm$ 0.097	0.401 $\pm$ 0.087	0.329 $\pm$ 0.088	0.390 $\pm$ 0.078	0.238 $\pm$ 0.087	0.296 $\pm$ 0.066	0.339 $\pm$ 0.055	0.358 $\pm$ 0.026	0.371 $\pm$ 0.015
(RCPS / e.m.c.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.831 $\pm$ 0.010
(RCPS / e.m.c.) Hoeffing-Bentkus	0.769 $\pm$ 0.090	0.795 $\pm$ 0.125	0.909 $\pm$ 0.029	0.340 $\pm$ 0.178	0.033 $\pm$ 0.017	0.138 $\pm$ 0.102	0.215 $\pm$ 0.130	0.024 $\pm$ 0.020	0.037 $\pm$ 0.012	0.075 $\pm$ 0.013
(RCPS / e.m.c.) Bernoulli KL	0.247 $\pm$ 0.160	0.233 $\pm$ 0.155	0.173 $\pm$ 0.111	0.210 $\pm$ 0.100	0.169 $\pm$ 0.093	0.085 $\pm$ 0.033	0.134 $\pm$ 0.079	0.089 $\pm$ 0.022	0.109 $\pm$ 0.026	0.113 $\pm$ 0.009
(RCPS / m.c.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.919 $\pm$ 0.127
(RCPS / m.c.) Hoeffing-Bentkus	1.000 $\pm$ 0.000	0.919 $\pm$ 0.127	1.000 $\pm$ 0.000	0.604 $\pm$ 0.252	0.136 $\pm$ 0.152	0.072 $\pm$ 0.031	0.422 $\pm$ 0.247	0.052 $\pm$ 0.031	0.054 $\pm$ 0.019	0.143 $\pm$ 0.025
(RCPS / m.c.) Bernoulli KL	0.321 $\pm$ 0.232	0.342 $\pm$ 0.226	0.357 $\pm$ 0.220	0.305 $\pm$ 0.183	0.342 $\pm$ 0.225	0.168 $\pm$ 0.146	0.114 $\pm$ 0.034	0.121 $\pm$ 0.022	0.214 $\pm$ 0.139	0.133 $\pm$ 0.014
(RCPS / l.p.) Emp. Bernstein	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
(RCPS / l.p.) Hoeffing-Bentkus	0.738 $\pm$ 0.145	0.795 $\pm$ 0.107	0.901 $\pm$ 0.104	0.380 $\pm$ 0.163	0.161 $\pm$ 0.105	0.223 $\pm$ 0.101	0.269 $\pm$ 0.130	0.081 $\pm$ 0.061	0.146 $\pm$ 0.061	0.299 $\pm$ 0.039
(RCPS / l.p.) Bernoulli KL	0.320 $\pm$ 0.199	0.233 $\pm$ 0.109	0.227 $\pm$ 0.083	0.384 $\pm$ 0.037	0.242 $\pm$ 0.088	0.304 $\pm$ 0.081	0.309 $\pm$ 0.083	0.299 $\pm$ 0.068	0.318 $\pm$ 0.066	0.382 $\pm$ 0.005

Table 9: Dataset Temporal Sequences (temp. = 0 response): UPPER CONFIDENCE BOUNDS ON CALIBRATION SET. $\alpha = 0.1$

baseline / sample size	10	15	20	25	30	40	60	100	300	900	3200
(. / e.m.c.) Baseline	0.090 $\pm$ 0.016	0.060 $\pm$ 0.010	0.080 $\pm$ 0.013	0.080 $\pm$ 0.000	0.093 $\pm$ 0.007	0.092 $\pm$ 0.008	0.093 $\pm$ 0.006	0.095 $\pm$ 0.004	0.097 $\pm$ 0.001	0.096 $\pm$ 0.001	0.097 $\pm$ 0.001
(CRC / e.m.c.) Bound in expectation	0.091 $\pm$ 0.000	0.062 $\pm$ 0.000	0.095 $\pm$ 0.000	0.073 $\pm$ 0.006	0.094 $\pm$ 0.005	0.093 $\pm$ 0.005	0.090 $\pm$ 0.006	0.095 $\pm$ 0.003	0.097 $\pm$ 0.002	0.097 $\pm$ 0.002	0.095 $\pm$ 0.002
(. / m.c.) Baseline	0.070 $\pm$ 0.024	0.053 $\pm$ 0.014	0.080 $\pm$ 0.013	0.048 $\pm$ 0.018	0.093 $\pm$ 0.007	0.090 $\pm$ 0.006	0.085 $\pm$ 0.006	0.086 $\pm$ 0.006	0.083 $\pm$ 0.006	0.089 $\pm$ 0.003	0.088 $\pm$ 0.003
(CRC / m.c.) Bound in expectation	0.091 $\pm$ 0.000	0.062 $\pm$ 0.000	0.081 $\pm$ 0.011	0.069 $\pm$ 0.008	0.084 $\pm$ 0.008	0.080 $\pm$ 0.008	0.089 $\pm$ 0.007	0.088 $\pm$ 0.008	0.093 $\pm$ 0.003	0.090 $\pm$ 0.005	0.086 $\pm$ 0.002
(. / l.p.) Baseline	0.090 $\pm$ 0.016	0.053 $\pm$ 0.014	0.085 $\pm$ 0.012	0.068 $\pm$ 0.013	0.097 $\pm$ 0.005	0.090 $\pm$ 0.006	0.092 $\pm$ 0.006	0.091 $\pm$ 0.005	0.098 $\pm$ 0.001	0.099 $\pm$ 0.001	0.098 $\pm$ 0.001
(CRC / l.p.) Bound in expectation	0.091 $\pm$ 0.000	0.062 $\pm$ 0.000	0.095 $\pm$ 0.000	0.077 $\pm$ 0.000	0.094 $\pm$ 0.005	0.095 $\pm$ 0.004	0.092 $\pm$ 0.007	0.093 $\pm$ 0.003	0.096 $\pm$ 0.002	0.097 $\pm$ 0.001	0.099 $\pm$ 0.000
(RCPS / e.m.c.) Emp. Bernstein	15.153 $\pm$ 0.000	9.741 $\pm$ 0.000	7.178 $\pm$ 0.000	5.682 $\pm$ 0.000	4.703 $\pm$ 0.000	3.497 $\pm$ 0.000	2.311 $\pm$ 0.000	1.378 $\pm$ 0.000	0.456 $\pm$ 0.000	0.141 $\pm$ 0.011	0.098 $\pm$ 0.001
(RCPS / e.m.c.) Hoeffing-Bentkus	0.091 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.014 $\pm$ 0.015	0.069 $\pm$ 0.012	0.090 $\pm$ 0.003	0.091 $\pm$ 0.006	0.083 $\pm$ 0.006	0.092 $\pm$ 0.003
(RCPS / e.m.c.) Bernoulli KL	0.070 $\pm$ 0.024	0.047 $\pm$ 0.016	0.100 $\pm$ 0.000	0.080 $\pm$ 0.000	0.090 $\pm$ 0.008	0.092 $\pm$ 0.008	0.100 $\pm$ 0.000	0.100 $\pm$ 0.000	0.096 $\pm$ 0.003	0.098 $\pm$ 0.001	0.096 $\pm$ 0.002
(RCPS / m.c.) Emp. Bernstein	4.371 $\pm$ 0.000	2.810 $\pm$ 0.000	2.0								

Table 11: Dataset TriviaQA (temp. = 0 response): UPPER CONFIDENCE BOUNDS ON CALIBRATION SET. $\alpha = 0.2$

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	0.180 $\pm$ 0.021	0.187 $\pm$ 0.014	0.180 $\pm$ 0.017	0.180 $\pm$ 0.019	0.177 $\pm$ 0.017	0.192 $\pm$ 0.008	0.193 $\pm$ 0.006	0.196 $\pm$ 0.003	0.196 $\pm$ 0.001	0.197 $\pm$ 0.002
(CRC / e.m.c.) Bound in expectation	0.173 $\pm$ 0.014	0.169 $\pm$ 0.021	0.176 $\pm$ 0.011	0.188 $\pm$ 0.006	0.187 $\pm$ 0.007	0.190 $\pm$ 0.005	0.190 $\pm$ 0.006	0.196 $\pm$ 0.002	0.195 $\pm$ 0.002	0.197 $\pm$ 0.002
(. / m.c.) Baseline	0.150 $\pm$ 0.035	0.173 $\pm$ 0.017	0.160 $\pm$ 0.032	0.156 $\pm$ 0.033	0.123 $\pm$ 0.045	0.170 $\pm$ 0.031	0.175 $\pm$ 0.031	0.170 $\pm$ 0.031	0.186 $\pm$ 0.007	0.185 $\pm$ 0.007
(CRC / m.c.) Bound in expectation	0.145 $\pm$ 0.023	0.144 $\pm$ 0.029	0.110 $\pm$ 0.033	0.127 $\pm$ 0.038	0.097 $\pm$ 0.041	0.156 $\pm$ 0.035	0.133 $\pm$ 0.040	0.172 $\pm$ 0.029	0.183 $\pm$ 0.005	0.183 $\pm$ 0.007
(. / l.p.) Baseline	0.180 $\pm$ 0.021	0.187 $\pm$ 0.014	0.180 $\pm$ 0.017	0.180 $\pm$ 0.019	0.183 $\pm$ 0.016	0.192 $\pm$ 0.008	0.193 $\pm$ 0.008	0.196 $\pm$ 0.003	0.198 $\pm$ 0.001	0.193 $\pm$ 0.002
(CRC / l.p.) Bound in expectation	0.173 $\pm$ 0.014	0.163 $\pm$ 0.022	0.186 $\pm$ 0.007	0.192 $\pm$ 0.000	0.184 $\pm$ 0.008	0.193 $\pm$ 0.004	0.192 $\pm$ 0.005	0.196 $\pm$ 0.003	0.198 $\pm$ 0.002	0.196 $\pm$ 0.001
(RCPS / e.m.c.) Emp. Bernstein	15.153 $\pm$ 0.000	9.741 $\pm$ 0.000	7.178 $\pm$ 0.000	5.682 $\pm$ 0.000	4.703 $\pm$ 0.000	3.497 $\pm$ 0.000	2.311 $\pm$ 0.000	1.378 $\pm$ 0.000	0.456 $\pm$ 0.000	0.194 $\pm$ 0.002
(RCPS / e.m.c.) Hoeffing-Bentkus	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.169 $\pm$ 0.004	0.124 $\pm$ 0.018	0.096 $\pm$ 0.021	0.186 $\pm$ 0.001	0.176 $\pm$ 0.014	0.185 $\pm$ 0.001	0.189 $\pm$ 0.002
(RCPS / e.m.c.) Bernoulli KL	0.170 $\pm$ 0.033	0.187 $\pm$ 0.014	0.190 $\pm$ 0.010	0.200 $\pm$ 0.000	0.183 $\pm$ 0.012	0.198 $\pm$ 0.004	0.197 $\pm$ 0.003	0.198 $\pm$ 0.003	0.197 $\pm$ 0.002	0.197 $\pm$ 0.002
(RCPS / m.c.) Emp. Bernstein	4.371 $\pm$ 0.000	2.810 $\pm$ 0.000	2.070 $\pm$ 0.000	1.639 $\pm$ 0.000	1.356 $\pm$ 0.000	1.009 $\pm$ 0.000	0.667 $\pm$ 0.000	0.397 $\pm$ 0.000	0.083 $\pm$ 0.000	0.042 $\pm$ 0.027
(RCPS / m.c.) Hoeffing-Bentkus	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.063 $\pm$ 0.040	0.105 $\pm$ 0.023	0.094 $\pm$ 0.021	0.110 $\pm$ 0.047	0.164 $\pm$ 0.017	0.175 $\pm$ 0.006	0.171 $\pm$ 0.009
(RCPS / m.c.) Bernoulli KL	0.110 $\pm$ 0.049	0.120 $\pm$ 0.043	0.135 $\pm$ 0.047	0.148 $\pm$ 0.040	0.127 $\pm$ 0.044	0.172 $\pm$ 0.031	0.193 $\pm$ 0.008	0.190 $\pm$ 0.006	0.168 $\pm$ 0.029	0.188 $\pm$ 0.007
(RCPS / l.p.) Emp. Bernstein	18.747 $\pm$ 0.000	12.052 $\pm$ 0.000	8.880 $\pm$ 0.000	7.030 $\pm$ 0.000	5.818 $\pm$ 0.000	4.326 $\pm$ 0.000	2.860 $\pm$ 0.000	1.704 $\pm$ 0.000	0.564 $\pm$ 0.000	0.169 $\pm$ 0.000
(RCPS / l.p.) Hoeffing-Bentkus	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.152 $\pm$ 0.027	0.124 $\pm$ 0.018	0.096 $\pm$ 0.021	0.182 $\pm$ 0.006	0.176 $\pm$ 0.014	0.182 $\pm$ 0.003	0.188 $\pm$ 0.004
(RCPS / l.p.) Bernoulli KL	0.110 $\pm$ 0.043	0.193 $\pm$ 0.010	0.195 $\pm$ 0.008	0.196 $\pm$ 0.006	0.183 $\pm$ 0.012	0.195 $\pm$ 0.005	0.197 $\pm$ 0.003	0.198 $\pm$ 0.002	0.196 $\pm$ 0.002	0.195 $\pm$ 0.002

Table 12: Dataset TriviaQA (temp. = 0 response): MEDIAN λ values. $\alpha = 0.2$

baseline / sample size	10	15	20	25	30	40	60	100	300	800
(. / e.m.c.) Baseline	1.322 $\pm$ 1.395	1.879 $\pm$ 1.286	0.0 $\pm$ 0.845	0.974 $\pm$ 1.101	2.715 $\pm$ 1.210	2.993 $\pm$ 1.011	3.829 $\pm$ 0.725	4.037 $\pm$ 0.875	4.177 $\pm$ 0.437	4.107 $\pm$ 0.237
(CRC / e.m.c.) Bound in expectation	2.854 $\pm$ 1.194	3.132 $\pm$ 1.034	5.430 $\pm$ 0.938	5.430 $\pm$ 1.194	5.082 $\pm$ 1.314	4.594 $\pm$ 0.890	4.734 $\pm$ 0.904	3.620 $\pm$ 0.489	4.107 $\pm$ 0.288	4.246 $\pm$ 0.279
(. / m.c.) Baseline	5.0 $\pm$ 2.352	5.5 $\pm$ 1.890	1.0 $\pm$ 2.112	2.0 $\pm$ 1.814	5.0 $\pm$ 2.112	5.0 $\pm$ 1.949	9.5 $\pm$ 1.296	9.5 $\pm$ 1.586	9.0 $\pm$ 0.666	9.0 $\pm$ 0.280
(CRC / m.c.) Bound in expectation	5.5 $\pm$ 1.907	7.0 $\pm$ 1.820	10.5 $\pm$ 1.511	10.0 $\pm$ 2.047	11.0 $\pm$ 1.980	8.0 $\pm$ 1.615	10.0 $\pm$ 1.468	8.5 $\pm$ 0.999	9.0 $\pm$ 0.364	9.0 $\pm$ 0.389
(. / l.p.) Baseline	-53.030 $\pm$ 13.363	-28.989 $\pm$ 13.659	-70.0 $\pm$ 12.545	-70.0 $\pm$ 13.425	-24.393 $\pm$ 12.151	-24.393 $\pm$ 11.315	-25.454 $\pm$ 3.385	-19.797 $\pm$ 8.184	-20.858 $\pm$ 1.276	-20.505 $\pm$ 0.396
(CRC / l.p.) Bound in expectation	-12.727 $\pm$ 10.807	-19.444 $\pm$ 11.023	-14.141 $\pm$ 9.917	-18.737 $\pm$ 11.476	-18.383 $\pm$ 9.234	-22.979 $\pm$ 7.986	-22.626 $\pm$ 7.811	-19.797 $\pm$ 2.529	-20.151 $\pm$ 1.113	-20.151 $\pm$ 0.871
(RCPS / e.m.c.) Emp. Bernstein	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.000	6.544 $\pm$ 0.032
(RCPS / e.m.c.) Hoeffing-Bentkus	6.544 $\pm$ 0.319	6.544 $\pm$ 0.443	6.822 $\pm$ 0.231	5.012 $\pm$ 1.220	1.949 $\pm$ 0.675	2.088 $\pm$ 1.074	3.132 $\pm$ 1.417	0.556 $\pm$ 0.743	2.367 $\pm$ 0.521	3.272 $\pm$ 0.267
(RCPS / e.m.c.) Bernoulli KL	0.626 $\pm$ 1.487	3.202 $\pm$ 1.208	3.550 $\pm$ 1.047	4.734 $\pm$ 0.647	4.594 $\pm$ 1.249	3.341 $\pm$ 0.834	3.620 $\pm$ 0.744	3.689 $\pm$ 0.632	4.316 $\pm$ 0.550	4.177 $\pm$ 0.217
(RCPS / m.c.) Emp. Bernstein	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	20.0 $\pm$ 0.000	11.0 $\pm$ 0.000	11.0 $\pm$ 0.156
(RCPS / m.c.) Hoeffing-Bentkus	11.0 $\pm$ 0.000	11.0 $\pm$ 0.156	11.0 $\pm$ 0.000	11.0 $\pm$ 2.197	4.0 $\pm$ 1.769	7.0 $\pm$ 1.574	7.0 $\pm$ 2.438	4.0 $\pm$ 1.722	6.0 $\pm$ 1.052	9.5 $\pm$ 0.635
(RCPS / m.c.) Bernoulli KL	4.5 $\pm$ 2.387	8.5 $\pm$ 2.195	8.5 $\pm$ 1.551	10.0 $\pm$ 0.944	8.5 $\pm$ 2.119	8.0 $\pm$ 1.254	8.5 $\pm$ 1.116	9.0 $\pm$ 0.700	9.0 $\pm$ 0.853	9.0 $\pm$ 0.208
(RCPS / l.p.) Emp. Bernstein	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000	0.0 $\pm$ 0.000
(RCPS / l.p.) Hoeffing-Bentkus	-0.353 $\pm$ 4.691	-0.353 $\pm$ 0.435	0.0 $\pm$ 1.651	-11.313 $\pm$ 13.548	-33.232 $\pm$ 12.190	-26.161 $\pm$ 8.869	-19.444 $\pm$ 14.608	-51.616 $\pm$ 11.280	-26.868 $\pm$ 7.022	-22.272 $\pm$ 1.040
(RCPS / l.p.) Bernoulli KL	-46.666 $\pm$ 16.077	-24.040 $\pm$ 13.039	-24.747 $\pm$ 8.076	-17.676 $\pm$ 1.981	-21.565 $\pm$ 11.774	-19.797 $\pm$ 3.794	-20.505 $\pm$ 4.256	-21.212 $\pm$ 2.377	-19.797 $\pm$ 2.051	-19.797 $\pm$ 0.275