ADDRESSING PITFALLS IN THE EVALUATION OF UNCERTAINTY ESTIMATION METHODS FOR NATURAL LANGUAGE GENERATION

Mykyta Ielanskyi[1], **Kajetan Schweighofer**[1], **Lukas Aichberger**[1], **Sepp Hochreiter**[1,2]
[1] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[2] NXAI GmbH, Linz, Austria
{ielanskyi, schweighofer, aichberger, hochreit}@ml.jku.at

ABSTRACT

Hallucinations are a common issue that undermine the reliability of large language models (LLMs). Recent studies have identified a specific subset of hallucinations, known as confabulations, which arise due to predictive uncertainty of LLMs. To detect confabulations, various methods for estimating predictive uncertainty in natural language generation (NLG) have been developed. These methods are typically evaluated by correlating uncertainty estimates with the correctness of generated text, with question-answering (QA) datasets serving as the standard benchmark. However, evaluating correctness in QA tasks is inherently challenging and can distort the perceived effectiveness of uncertainty estimation methods. Our results show that there is substantial disagreement between correctness functions and consequently the ranking of the uncertainty estimation methods is significantly influenced by that choice, allowing to inflate the performance of uncertainty estimation methods. We propose several alternative risk indicators for correlation assessment that improve robustness of empirical assessment of UE algorithms for NLG. For QA tasks, we show that averaging over multiple LLM-as-a-judge variants leads to more reliable results. Furthermore, we explore structured tasks which provide unambiguous correctness functions. Finally, we propose to use an Elo rating of uncertainty estimation methods to give an objective summarization over extensive evaluation settings.

1 INTRODUCTION

Predictive uncertainty has been linked to the occurrence of a subset of hallucinations known as confabulations (Farquhar et al., 2024). Such confabulations are sequences generated by a large language model (LLM), that have no support in either the training set of the model nor in the prompt. The expressivity of natural language allows these models to obfuscate their lack of knowledge in a manner that can be challenging to detect. Therefore, uncertainty estimation is essential to detect such confabulations and ensure the reliability and wider applicability of LLM-based systems.

Predictive uncertainty in natural language generation (NLG) can be quantified by the entropy of the LLMs predictive distribution Malinin & Gales (2020). In the literature on uncertainty estimation in univariate classification, predictive uncertainty is often decomposed into aleatoric and epistemic components Gal (2016). The aleatoric uncertainty can be attributed to the inherent stochasticity of the prediction, while the epistemic uncertainty arises from lack of knowledge of the true model parameters Schweighofer et al. (2023). In case of confabulation detection in NLG, most of the time the aleatoric uncertainty of predicting with a given model with parameters $\boldsymbol{w}$ for a new input $\boldsymbol{x}$ is considered Aichberger et al. (2024b).

Currently, uncertainty estimation algorithms for NLG are evaluated mostly in terms of selective prediction on a narrow class of problems which is question-answering (QA) datasets, see Table 1. These datasets require models to either retrieve factual information from their weights (closed book QA) or a given prompt (open book QA). The motivation for using QA tasks is that the ability to

Table 1: Evaluation protocols recently used for uncertainty estimation in NLG. Few works evaluate their methods beyond selective prediction on QA tasks and rely on approximate correctness functions or a small number of human correctness evaluations.

REFERENCE	TASK	CORRECTNESS
MALININ & GALES (2020)	TRANS.	BLEU
FOMICHEVA ET AL. (2020)	TRANS.	HUMAN
KUHN ET AL. (2023)	QA	ROUGE-X
FADEEVA ET AL. (2023)	QA, SUM.	ROUGE-X, BERTSCORE
DUAN ET AL. (2024)	QA	ROUGE-X
MANAKUL ET AL. (2023)	FACT.	HUMAN
FARQUHAR ET AL. (2024)	QA	JUDGE
BAKMAN ET AL. (2024)	QA	JUDGE
AICHBERGER ET AL. (2024B)	QA	ROUGE-X, BLEURT
CHEN ET AL. (2024)	QA	ROUGE-X
KOSSEN ET AL. (2024)	QA	JUDGE, F-1
NIKITIN ET AL. (2024)	QA	JUDGE
AICHBERGER ET AL. (2024A)	QA	JUDGE, F-1
ABBASI-YADKORI ET AL. (2024)	QA	F-1

effectively retrieve information could be linked to factuality and hallucinations and has a relatively low demand for model performance. However, this class of problems is characterized by short length of the expected answer and impreciseness of the ground truth solution. Importantly, the evaluation of an answer is done by approximate correctness functions, such as comparing substrings or utilizing text similarity models. These correctness functions have been criticized and are often not considered robust (Schluter, 2017; Zheng et al., 2023; Santilli et al., 2024), yet are widely used in NLG. Specifically, Santilli et al. (2024) investigates the relation between the Rouge-L, LLM-as-a-judge and Human annotators and the impact it has on the empirical performances reported in (Farquhar et al., 2024) (Fadeeva et al., 2023). They conclude that LLM-as-a-judge should be preferred as a correctness metric in such assessments and the effects of thresholds should further be estimated and sequence length is an important factor in variability of outcomes. At the same time, Zheng et al. (2023), the original work proposing LLM-as-a-judge approach, already point out biases inherent to the approach. We conduct a more detailed investigation of the effects that the approximate correctness has on the ranking of the NLG uncertainty estimation algorithms and propose improvements.

Our contributions are as follows:

- We conduct a detailed investigation of weaknesses of the evaluation practices used in recent work on uncertainty estimation in NLG, some of which have been pointed out in prior work.
- We suggest several alternative risk indicators to be used for correlation experiments. In particular, we suggest assessing correctness with an ensemble of LLM-as-a-judge variants for QA, structured tasks with exact correctness functions, OOD detection and perturbation.
- We propose an Elo rating based aggregation for comparing the performance of uncertainty estimation methods across different experimental setups to foster a more objective assessment of their utility.

2 PRELIMINARIES

The uncertainty estimation problem in NLG can be formalized as follows: given an input sequence $\boldsymbol{x} = (x_1, ..., x_\tau) \in \mathcal{X}$ and a model with parameters $\boldsymbol{w}$, we want to infer an uncertainty measure $u : \mathcal{X} \times \mathcal{W} \mapsto R$. Then $\hat{u}(\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}_u)$ is an algorithm to obtain an estimate of $u(\boldsymbol{x}, \boldsymbol{w})$, where $\boldsymbol{\theta}_u$ is a vector of hyperparameters of the algorithm.

Uncertainty estimation methods in NLG. Recent approaches to uncertainty estimation for NLG estimate uncertainty in a variety of ways. The methods can be loosely categorized into three groups: those using statistics of a set of sequences from the model, those using a single output sequence and those using heuristics. The first group of methods is based on Monte-Carlo (MC) sampling and

Bayesian assumptions with regard to the obtained samples. Such methods base their estimators on some notion of spread in the probability space of the predictive distribution of an LLM (Malinin & Gales, 2020; Kuhn et al., 2023; Aichberger et al., 2024b; Chen et al., 2024; Nikitin et al., 2024). A noteworthy variation of this direction consists of methods that attempt to modify the probabilities of sampled sequences based on the semantic importance of individually generated subsequences (Duan et al., 2024; Bakman et al., 2024) to compensate for the potential impact that they make on the correctness of the predicted sequence. The approaches from the second group use properties of a single generated sequence (Ren et al., 2023; Fadeeva et al., 2023; Kossen et al., 2024; Aichberger et al., 2024a) to estimate the model's confidence. Heuristic approaches leverage the facilities of the language model itself or a larger one to determine confidence estimate for a given output sequence (Kadavath et al., 2022; Manakul et al., 2023). Detailed reference of the methods considered in this work can be found in Appendix A.1.

Evaluating uncertainty estimation. Intuitively, the fundamental question to which $u(\boldsymbol{x}, \boldsymbol{w})$ should help us find the answer is: "What is the risk of making the prediction for a given input sequence $\boldsymbol{x}$ using the model with parameters $\boldsymbol{w}$?". This connection of uncertainty and risk has recently been advocated in the univariate classification setting Lahlou et al. (2023); Kotelevskii & Panov (2025). In accordance with this perspective, the performance of uncertainty estimation methods is empirically evaluated as a correlation between the estimated uncertainty $\hat{u}(\cdot)$ and some risk indicator $r(\cdot)$ on sets of predictions, defined as

$$\xi \;=\; \mathit{Cor}\left[\left(\hat{u}(\boldsymbol{x}_i, \boldsymbol{w}; \boldsymbol{\theta}_u)\right), \left(r(\boldsymbol{x}_i, \boldsymbol{y}_i')\right)\right] \;. \tag{1}$$

Here, *Cor* is a correlation metric and $\boldsymbol{y}'$ is the predicted output sequence of the LLM. We do not assume a linear relation between the risk and the uncertainty, which restricts eligible *Cor* to rank correlation metrics, e.g. Spearman ρ, Area Under the ROC Curve (AUC), Area Under Precision-Recall Curve (AUPR). In this work we will not consider calibration of uncertainty values that may be performed to obtain a prediction risk classifier.

Selective prediction. The current standard for comparing uncertainty estimation methods for NLG is selective prediction on QA datasets (Aichberger et al., 2024b; Kuhn et al., 2023; Farquhar et al., 2024; Duan et al., 2024; Bakman et al., 2024). This approach uses a correctness indicator as a proxy for risk. The correctness function $c : \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \mapsto \{0, 1\}$ assigns prediction to be incorrect or correct. Then the selective prediction performance is defined as follows:

$$\xi_{\text{SP}} \;=\; \mathit{Cor}\left[\left(\hat{u}(\boldsymbol{x}_i, \boldsymbol{w}; \boldsymbol{\theta}_u)\right), \left(\neg c(\boldsymbol{y}_i', \boldsymbol{y}_i, \boldsymbol{x}_i; \boldsymbol{\theta}_c)\right)\right] \;. \tag{2}$$

Most commonly, AUC is used as correlation metric *Cor*, capturing the uncertainty scores ability to distinguish between correct and incorrect predictions. It represents the probability that a randomly chosen correct sample is ranked higher than a randomly chosen incorrect sample in terms of the uncertainty score. Note that the correctness function is assumed to be deterministic. Otherwise, additional sampling would be required to integrate out the randomness.

Standard correctness functions in NLP. The standard substring matching correctness algorithms are the ROUGE (Lin, 2004) and BLEU families (Papineni et al., 2002). These algorithms evaluate textual similarity on the n-gram level. To turn those functions into a correctness function, one is required to specify a threshold d and the n-gram parameter n, so $\boldsymbol{\theta}_c = (d, n)$. Learned correctness functions, such as BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020) use similarity of the answer and the reference in an embedding space. LLM-as-a-judge (Zheng et al., 2023) prompts an LLM to confirm the correctness of the answer with respect to the reference.

3 Pitfalls of Current Evaluation Procedure

In the univariate classification setting the correctness function is very simple, usually consisting of selecting the highest probability output class and checking its identity to the class provided as the label. In NLG, correctness algorithms are more complex due to the large space of possible sequences and certain degree of invariance to syntactic permutations and paraphrasis. Selective prediction performance (see Eq. (2)) depends on both the estimation quality of the uncertainty estimation method and the bias and variance of the correctness function used.

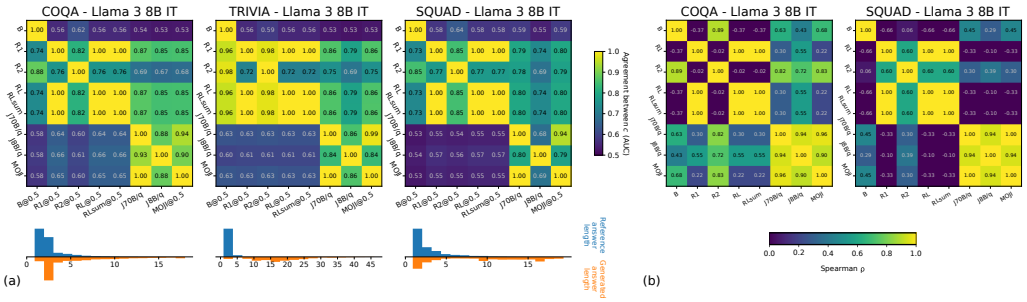

Figure 1: Correctness metrics on selected QA datasets. R indicates ROUGE family, B - BLEU. judge models are indicated with J, 'q' stands for QA prompt used in Farquhar et al. (2024) (see Sec. C for more details on prompting). **(a)** Agreement of correctness metrics. Rows are the predicting correctness function. Columns are the 'ground truth' values discretized at a threshold indicated by @. Higher values correspond to higher agreement between correctness assignment. Note that for the AUC score, the values in this matrix need not be symmetric. **(b)** Agreement between the ranking of uncertainty estimation algorithm that arises from given approximate correctness functions. ρ of 1 indicates identical ordering while ρ of 0 indicates uncorrelated rank assignment by the two correctness functions.

Effects of bias and variance in labels on the AUC estimation In Appx.E.2 we investigate the effects that bias and noise in correctness function have on AUC estimation. We conclude, that presence of either substantially increase the bias of the AUC estimator which depends on the degree of distortion of risk indicator labels. Sample independent Bernoulli modeled label noise affects all of the UE methods equally in the asymptotic case (Eq.(21)). At the same time, bias in correctness estimates affects the ranking proportionally to the a) proportion of distorted samples (Eq.(25)); b) the discrepancy in ranking quality on the distorted and undistorted samples (Eq.(24)).

From this follows, that if we do not sum out the random noise in the risk indicator it will turn into a sample-dependent distortion, which can affect the apparent performances of different methods differently. *Most prior work ignores the non deterministic nature of the correctness estimates used.* This is particularly relevant for LLM-as-a-judge approach since it utilizes a stochastic model.

Agreement of different correctness functions. In Fig. 1 (a) we compare the predictions of widely used correctness functions on the QA datasets commonly used for comparing NLG uncertainty estimation algorithms. We observe that the n-gram based correctness function families BLEU and ROUGE show substantial disagreement between each other and the LLM-as-a-judge. Different variants of ROUGE show high agreement among them in some scenarios. This agreement can be largely explained by short reference answers provided for the QA datasets, rendering these n-gram based metrics equivalent in most scenarios. As can be seen in the bottom part of Fig. 1 (a), the reference answer lengths are very short, with most consisting of only one or two words. This demonstrates bias and noise in approximate correctness estimators. Furthermore, during our investigation, we have discovered a prominent artifact in a widely used ROUGE-2 and BLEU implementation (Luong et al., 2017) described in more detail in Appx.A.3.

Consistency of uncertainty estimation method ranking. Fig. 1 (b) depicts Spearman correlations between the ranks of NLG uncertainty estimation algorithms evaluated on the given datasets according to different correctness functions. On both CoQA and SQuAD it can be observed that the disagreement in ranking uncertainty estimation methods falls on the lines between judge and n-gram methods with a noticeable BLEU / ROUGE-2 artifact. The adaptive ROUGE metrics are in perfect agreement with ROUGE-1 due to low length of reference answers. The judge models agree more with the BLEU / ROUGE-2 than with other ROUGE variants. This indicates, that among the approximate methods the LLM-as-a-judge might be the more reliable one, although not universally.

Correctness-hacking QA benchmarks. In Tab. 2 we show results of optimizing the performance of uncertainty estimation methods with respect to the correctness function. The experiment shows

Table 2: Adversarially selecting a correctness function on the QA benchmark to improve the ranking of individual uncertainty estimation methods. The values are frequencies of uncertainty estimation methods Top-3 membership on the considered QA datasets. Ref. is a judge average introduced in Sec. 4 and is a reference for performance assessment. Correctness function and thresholds which are not frequently used in the literature such as BLEU and ROUGE-2 were not made available for optimization (limiting thresholds to 0.3 and 0.5).

METHOD	REF.	ADVERSARIAL	INCREASE
PRED. ENT.	0.000	0.188	+0.188
PRED. ENT. (LN)	0.000	0.125	+0.125
SEQ. LEN. (SAMPLE)	0.250	0.312	+0.062
SEQ. LEN. (ANSWER)	0.312	0.562	+0.250
EIGENSCORE	0.125	0.250	+0.125
TOKENSAR	0.062	0.062	+0.000
SENTENCESAR	0.438	0.556	+0.118
SAR	0.125	0.188	+0.062
PERPLEXITY	0.125	0.444	+0.319
MIN TOK. LOG PROB.	0.125	0.500	+0.375
SEMANTIC ENT.	0.125	0.333	+0.208
SEMANTIC ENT. (LN)	0.562	0.667	+0.104
P(TRUE)	0.250	0.375	+0.125
G-NLL	0.375	0.688	+0.312

that the apparent performance of the methods can often be improved substantially compared to the value obtained for $\tilde{c}_{\text{reference}}$ by selecting an opportune correctness function $\tilde{c}$ and parametrization θ_c. This also holds for some of the introduced heuristic uncertainty measures, like the sequence length of the most likely generation. This way, if we were to propose e.g. the length of the generated output sequence as a novel method of uncertainty estimation for NLG we could provide plausible empirical support for it by adjusting the evaluation settings within reasonable bounds. This further highlights the vulnerabilities of the current evaluation strategy.

4 IMPROVING THE EVALUATION WITH ROBUST RISK INDICATORS

To alleviate the pitfalls we have demonstrated in the previous section, we propose several robust risk indicators to improve evaluation reliability. In the univariate classification setting, the following risk indicators are considered: correctness of the prediction, out-of-distribution (OOD) identifier and perturbation strength Gal & Ghahramani (2016); Lakshminarayanan et al. (2017); Malinin & Gales (2018); D' Angelo & Fortuin (2021); Daxberger et al. (2021); Mukhoti et al. (2023); Schweighofer et al. (2023). We refer to the general assumptions used for assessment of UE algorithms in Appx.E.1.

In this section, we first aim to obtain a more stable approximate correctness function $\tilde{c}$ for QA datasets. Second, we investigate code generation and constrained text generation, as those structured tasks have exact correctness functions. Finally, we discuss alternative risk indicators in out-of-distribution and perturbation detection.

Reducing variability of approximate correctness. Judge models are subject to biases and uncertainty with respect to sampling (Zheng et al., 2023). This variation is in part caused by the aleatoric and epistemic uncertainties of the judge model. To mitigate these effects, we propose using a Mixture of judges and Instructions (MoJI) to evaluate correctness on QA datasets. MoJI follows up on the possibility to sample stochastic correctness functions in Eq. (2). At the core of this approach lies integrating out the variability that arises from sampling the judge models, the architecture of the judge model and the prompt used as per Sec.3. MoJI approach considers averaging multiple predictions by multiple models with multiple prompts. To further improve robustness of the downstream uncertainty method evaluation, we could exclude all the examples, for which the predictive entropy of MoJI is above a predefined threshold. This would remove the correctness predictions with the highest judge disagreement. We evaluate MoJI on structured tasks, where exact correctness functions are available as a ground-truth. The specific datasets (BCB & COLLIE) will be introduced in

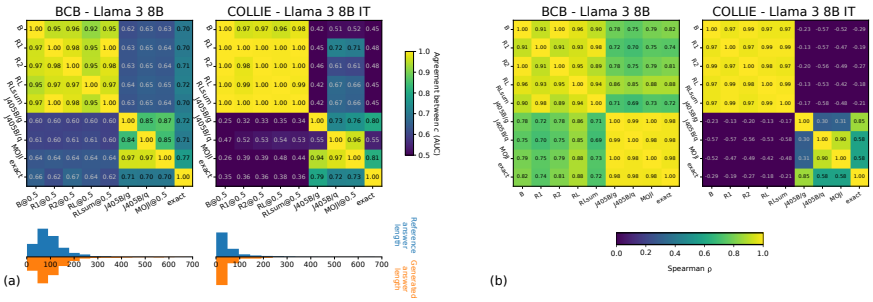

Figure 2: Correctness metrics on selected structured datasets. R indicates ROUGE family, B - BLEU. judge models are indicated with J, 'q' stands for QA prompt used in Farquhar et al. (2024) while 'g' stands for a more general prompt to evaluate correctness (see Apx. C for more details on prompting). **(a)** Agreement of correctness metrics. Rows are the predicting correctness function. Columns are the 'ground truth' values discretized at a threshold indicated by @. Higher values correspond to higher agreement between correctness assignment. **(b)** Agreement between the ranking of uncertainty estimation algorithm that arises from approximate and exact correctness functions.

detail in the next paragraph. Results are shown in and Fig. 2 (a), showing that MoJI agrees most with exact correctness.

Exact correctness. If we use problems with non-parametric correctness function c_e, we can improve the reliability of evaluating uncertainty estimation algorithms. We refer to this unambiguous non-parametric correctness function as *exact correctness*. Practical tasks where such correctness is defined would be problems that are non trivial to solve but can be verified symbolically. Such tasks are often called *structured problems*. We set our goal to avoid generating any new datasets, but rather to select suitable existing datasets. Specifically, we investigate code generation and constrained text generation tasks, which feature exact correctness functions.

The correctness space in *code completion* is very explicitly defined as the subset of sequences that can be compiled into a program correctly and pass a predefined suite of unit tests. The correctness function is then the binary label describing the fulfillment of all of the unit tests. Such correctness function is non parametric. There are several popular public datasets for code completion (Austin et al., 2021; Chen et al., 2021; Hendrycks et al., 2021; Li et al., 2022) that feature acceptable test coverage rate. BigCodeBench (BCB) Zhou et al. (2023) focuses on more applied aspects of python programming. Python is prevalent in the training data of modern LLMs and we therefore select BCB for our experiments.

Constrained text generation refers to generating a coherent passage of text that fulfills some specific and measurable requirements. E.g. producing a paragraph with three sentences such, that the last word of the second sentence is "uncertainty". This allows for automatic non-parametric correctness checking of the output for generations multiple paragraphs long. COLLIE (Yao et al., 2024) is a dataset and evaluation pipeline, focusing on constrained text generation. The evaluation of the answers is deterministic, defined by a list of symbolic constraints. This allows for fine grained control of both the query string and the correctness function. The problems in the dataset were extracted from curated text corpora assuring that a solution satisfying the symbolic constraints exists. This dataset is frequently used to assess the reasoning abilities of language models as it contains subtasks considered difficult for transformer-based models such as e.g. counting words and characters.

Results on BCB and COLLIE. Fig. 2 (a) shows the agreement between the correctness metrics on structured tasks. As was pointed out before, MoJI is in greatest agreement with the exact correctness in both cases. The generation and reference answer lengths seen on the bottom of the figure are an order of magnitude larger than those on the QA datasets. Fig. 2 (b) shows the Spearman correlation between ranks of uncertainty estimation methods. The approximate correctness functions struggle for the COLLIE dataset, with only the largest models with specific prompt being correlated to exact correctness. This is due to the fact, that the reference sequence may have completely different semantics compared to the one generated by the LLM, while both fulfilling the specified require-

ments. LLM-as-a-judge fails to pick this up unless the prompt is adopted to the structured tasks. While other structured tasks could be suitable for this evaluation, we consider these two datasets for their convenience.

OOD label as risk indicator. As pointed out earlier, incorrectness of prediction on in distribution data is not the only risk identifier used in uncertainty literature. In classification setting OOD detection and perturbation detection tasks are used alongside selective prediction (Appx.E.1). In these tasks it is assumed, that the risk of using the model on an input that violates the i.i.d. assumption or is corrupted is higher than that of an in distribution example. The evaluation would then, of all thing, depend on the quality of the OOD identifier $o : \mathcal{X} \mapsto \{0, 1\}$, resulting in the performance measure:

$$\xi_{\text{OOD}} \;=\; \mathit{Cor}\left[\left(\hat{u}(\boldsymbol{x}_i, \boldsymbol{w}; \boldsymbol{\theta}_u)\right), \left(o(\boldsymbol{x}_i)\right)\right] \tag{3}$$

Unlike the image domain, obtaining OOD examples for text data is difficult. When thinking of OOD examples for text, one would imagine questions about things that have not yet come to be or are otherwise unknown or ambiguous in the general text corpora. Several datasets seek to provide artificial OOD examples. Known-Unknowns Amayuelas et al. (2024) seek to collect questions that can be assumed to have controversial answers in the common training sets. SQuADv2 Rajpurkar et al. (2018) provides questions formulated to be unanswerable given the prompt. We consider these datasets in the experiments in the next section.

5 Aggregating Benchmark Results

Once a reliable risk indicator is selected, we are presented with quantitative assessments of uncertainty estimation methods for each model, dataset and sampling parameter considered. It is commonplace to see large tables listing every feasible combination of aforementioned factors which often feature contradicting assessments depending on e.g. the model or datasets used. Depending on how different results are highlighted or how these are discussed in the text, different conclusions, sometimes conflicting, can be drawn from the same raw results. To the extent of our knowledge, no work on uncertainty estimation in NLG has made an attempt to aggregate all of the experimental information from testing under diverse datasets and models into a single scalar in a grounded fashion.

Elo rating of uncertainty estimation methods. Drawing inspiration from popular approaches used for general evaluation of LLM skills (Chiang et al., 2024), we use the Elo rating system (Elo, 1978) to gain high level insight into performance of the considered uncertainty estimation methods. Originally intended to rate skill of chess players, the Elo rating provides an iterative algorithm to compute relative performance of players based on pairwise comparisons (games). We will treat each independent dataset / model risk correlation experiment (Eq. (1)) as a separate game, where the players, methods A and B, can win the game by having higher performance according to Eq. (2) or Eq. (3). The pairs and experimental runs are then sampled uniformly until the ratings converge to a stationary distribution that is defined by their relative per problem performance (Cortez & Tossounian, 2024). While each prediction in each considered dataset could be considered a separate game when estimating the Elo ratings, for the sake of consistency and to avoid unnecessary additional complexity we only consider the outcomes of experiments on the full datasets.

One advantage of the Elo system is the probabilistic interpretability of the scores. With the usual initialization, 400 point difference roughly corresponds to $1 : 10$ chances of one method being better than another for a model / dataset combination. Another advantage is it enables for indirect comparisons. E.g. if UE methods are evaluated on only partially overlapping sets of tasks, we could still aggregate their relative performance. This is not straightforward with rank based aggregation (e.g. Vashurin et al. (2025)). Finally, Elo score naturally accommodates variability of outcomes within the same experiment as well as allows prioritizing specific subsets of experiments.

The Elo ratings are presented in Fig. 3. The 'all task' ratings are the summary rating from which to draw conclusions about the general performance of the uncertainty estimation methods. It appears that the characteristics required to excel in different partitions of the experimental suite vary. Employing an effective aggregation approach allows us to gain new insights into comparative perfor-

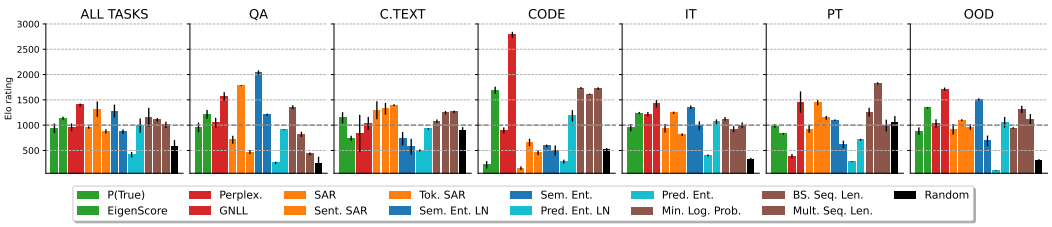

Figure 3: Elo ratings of NLG uncertainty estimation methods. The methods are grouped by color according to their category (see Apx. A.1). The 1000 line indicates the average rating. Elo rating were independently estimated for several key partitions. Per task used: QA - restricted to selective prediction on QA datasets, C.TEXT - to constrained text generation, CODE - to code completion. Per models used: IT - instruction fine tuned models only, PT - pretrained models only. OOD - OOD detection tasks only.

mance of the uncertainty estimation methods as well as to confirm some of the side note observations made in prior work.

6 Results and Conclusion

By applying our methodology, we attain several insights into relative performances of the uncertainty estimation methods for NLG. One key insight is that simple heuristic methods have competitive performance in most settings. We also observe that most of the time, length normalization is harming the performance of the uncertainty estimation methods. The only method where it improves the performance is where the information about the sequence likelihood is largely redundant (semantic entropy). Adjusting the likelihoods of individual tokens based on semantic importance measures appears to be counterproductive in most situations. Methods that utilize the representations of the generating model may require intricate, task-specific hyperparameter tuning to match the performance of information-theoretic methods.

Semantic entropy and sentence SAR are the top performers on the QA partition of the experiments. Applying Token level shifting appears to not yield performance benefits despite being computationally more expensive than the initial rollout of the predicting LLM in case of long generations. Length normalization appears to consistently hurt the performance of predictive entropy and consistently improve that of semantic entropy. This is peculiar, as length normalization of sequence likelihood was initially introduced as a feature meant to improve the performance of predictive entropy (Malinin & Gales, 2020). This could also explain good performance of discrete semantic entropy (Farquhar et al., 2024), where the sequence likelihoods are ignored entirely and the entropy is computed in the semantic cluster size space.

Both P(True) and EigenScore do not excel in any of the tasks in particular. This could be a result of the considered models that we have evaluated, as well as intricacies of prompting for P(True) and hyperparameter tuning for EigenScore (selecting the layer with the most suitable representations for a given task). The efficient G-NLL method (Aichberger et al., 2024a) shows consistently better than average performance across settings, particularly excelling at code completion and OOD detection. The ratings on IT and PT partitions indicate, that PT models should be avoided in these benchmarks, at the very least with problems that imply a query-response format rather than pure completion. Perplexity got rated above average on the IT setting, which might be a somewhat more realistic assessment than 'all tasks', but there is no specific task at which it could act as the go to choice.

Overall, our results suggest that there is no one-size-fits-all in uncertainty estimation for NLG, with different tasks having different method preferences. We expect that our analysis and suggestions will improve the standards for the evaluation of NLG uncertainty estimation algorithms and raise awareness about the caveats of the currently prevailing protocols.

7 ACKNOWLEDGMENTS

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. This research was funded in part by the Austrian Science Fund (FWF) [10.55776/COE12]. We thank the projects INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA (FFG-892171), FWF AIRI FG 9-N (10.55776/FG9), AI4GreenHeatingGrids (FFG- 899943), INTEGRATE (FFG-892418), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01). We thank NXAI GmbH, Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo), Software Competence Center Hagenberg GmbH, Borealis AG, TÜV Austria, Frauscher Sensonic, TRUMPF and the NVIDIA Corporation.

REFERENCES

Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterative prompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, and et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024.

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking Uncertainty Estimation in Natural Language Generation. *arXiv*, 2412.15176, 2024a.

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Semantically Diverse Language Generation for Uncertainty Estimation in Language Models. *arXiv*, 2406.04306, 2024b.

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Yang Wang. Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6416–6432, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program Synthesis with Large Language Models. *arXiv*, 2108.07732, 2021.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. MARS: Meaning-Aware Response Scoring for Uncertainty Estimation in Generative LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7752–7767, Bangkok, Thailand, 2024.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob

McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *arXiv*, 2107.03374, 2021.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Roberto Cortez and Hagop Tossounian. Convergence and stationary distribution of Elo rating systems. *arXiv*, 2410.09180, 2024.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN 0471241954.

Francesco D' Angelo and Vincent Fortuin. Repulsive Deep Ensembles are Bayesian. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3451–3465, 2021.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace Redux - Effortless Bayesian Deep Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20089–20103, 2021.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and Aurelien Rodriguez et al. The llama 3 herd of models. *arXiv*, 2407.21783, 2024.

A.E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Publishing Inc., 1978. ISBN 978-0-668-04721-0.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-polygraph: Uncertainty estimation for language models. In Yansong Feng and Els Lefever (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461. Association for Computational Linguistics, 2023.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8: 539–555, 09 2020.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8175–8195. PMLR, 17–23 Jul 2022.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language Models (Mostly) Know What They Know. *arXiv*, 2207.05221, 2022.

Andreas Kirsch. *Advancing Deep Active Learning & Data Subset Selection: Unifying Principles with Information-Theory Intuitions*. PhD thesis, University of Oxford, 2024.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs. *arXiv*, 2406.15927, 2024.

Nikita Kotelevskii and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via bayesian estimation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*, 2017.

Carsten Tim Lüth, Till J. Bungert, Lukas Klein, and Paul F. Jaeger. Navigating the Pitfalls of Active Learning Evaluation: A Systematic Framework for Meaningful Performance Assessment. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Andrey Malinin and Mark Gales. Uncertainty Estimation in Autoregressive Structured Prediction. In *International Conference on Learning Representations*, 2020.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfcheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deep Deterministic Uncertainty: A New Simple Baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24384–24394, 2023.

Alexander V Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A Method for Automatic Evaluation of Machine Translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, 2002.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, 2018.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *arXiv*, 1808.07042, 2019.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Andrea Santilli, Miao Xiong, Michael Kirchhof, Pau Rodriguez, Federico Danieli, Xavier Suau, Luca Zappella, Sinead Williamson, and Adam Golinski. On a Spurious Interaction between Uncertainty Scores and Answer Evaluation Metrics in Generative QA Tasks. In *Neurips Safe Generative AI Workshop 2024*, 2024.

Natalie Schluter. The limits of automatic summarisation according to ROUGE. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 41–45, Valencia, Spain, 2017.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of Uncertainty with Adversarial Models. *Advances in Neural Information Processing Systems*, 36:19446–19484, 2023.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning Robust Metrics for Text Generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, Online, 2020.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. Benchmarking Uncertainty Quantification Methods for Large Language Models with LM-Polygraph. *arXiv*, 2406.15627, 2025.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4697–4708. Curran Associates, Inc., 2020.

Shunyu Yao, Howard Chen, Austin W. Hanjie, Runzhe Yang, and Karthik R Narasimhan. COLLIE: Systematic construction of constrained text generation tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT, 2020.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5506–5524, Singapore, December 2023. Association for Computational Linguistics.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7B language model, 2024.

A Datasets and Methods Used In Our Analysis

A.1 Considered Uncertainty Estimation Methods

In the following section, we give an overview about the considered uncertainty estimation methods. Two main categories are methods that operate on multiple and single generated output sequences. Furthermore, an inherent problem of generating output sequences of arbitrary size (though in practice often capped as a maximum length), introduces the problem of having an uncertainty estimate that is independent of the sequence length. For methods based on output probabilities $p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$, this usually involves non-uniform weighting of individual token probabilities. Finally, we present a set of well performing heuristics.

A.1.1 Multiple Output Sequences

Many works investigate measures of sequence-level uncertainty that are defined as expectations over the sequence probability distribution $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ under a given model. Monte-Carlo (MC) approximations thereof rely on sampling multiple output sequences.

Predictive Entropy. Similar to the univariate classification setting, Predictive Entropy (Malinin & Gales, 2020) captures the variability in possible outcome sequences. If PE is high, the language model is likely to generate different outcome sequences. However, as the language model does not provide the full predictive distribution $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$, but only the conditional distribution $p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$ for each token. Therefore, estimating the Predictive Entropy necessitates a MC approximation:

$$\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})) \;=\; \mathrm{E}_{p(\boldsymbol{y}\mid\boldsymbol{x},\boldsymbol{w})}\left[-\log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\right] \;\approx\; \frac{1}{N}\sum_{n=1}^{N} -\log p(\boldsymbol{y}^n \mid \boldsymbol{x}, \boldsymbol{w})\,, \qquad \boldsymbol{y}^n \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\,. \tag{4}$$

Semantic Entropy. Predictive Entropy does not account for the fact that output sequences $\boldsymbol{y}$ are different, yet covey the same semantics. For example, "John is my brother." and "My brother is John" is semantically equivalent, yet are different output sequences. To that end, Kuhn et al. (2023); Farquhar et al. (2024) introduce Semantic Entropy, which accounts for those semantic equivalences. They do so by introducing a semantic cluster probability $p(c \mid \boldsymbol{x}, \boldsymbol{w})$, that is marginalized over possible output sequences:

$$p(c \mid \boldsymbol{x}, \boldsymbol{w}) \;=\; \sum_{\mathcal{Y}} p(c \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \tag{5}$$

In practice, Kuhn et al. (2023); Farquhar et al. (2024) suggest to deterministically assign output sequences to clusters. Semantic Entropy (Kuhn et al., 2023; Farquhar et al., 2024) is then defined on this cluster probability distribution:

$$\mathrm{H}(p(c \mid \boldsymbol{x}, \boldsymbol{w})) \;=\; \mathrm{E}_{p(c\mid\boldsymbol{x},\boldsymbol{w})}\left[-\log p(c \mid \boldsymbol{x}, \boldsymbol{w})\right] \;\approx\; \frac{1}{N}\sum_{n=1}^{N} -\log p(c^n \mid \boldsymbol{x}, \boldsymbol{w})\,, \qquad c^n \sim p(c \mid \boldsymbol{x}, \boldsymbol{w})\,. \tag{6}$$

Note that while the MC estimate in Eq. (6) is possible if one has access to the cluster probability distribution, this is not the case in practice. Therefore, we use the implementation of Aichberger et al. (2024b), who discuss how to construct a proper MC estimator of Semantic Entropy.

SentenceSAR. Instead of clustering, Duan et al. (2024) propose to add a consistency dependent penalty to the uncertainty calculation. The resulting measure, SentenceSAR is defined as

$$\text{SentenceSAR} \;=\; \frac{1}{N}\sum_{n=1}^{N} -\log p(\boldsymbol{y}^n \mid \boldsymbol{x}, \boldsymbol{w}) \;+\; \frac{\sum_{k\neq n} sim(\boldsymbol{y}^n, \boldsymbol{y}^k)\, p(\boldsymbol{y}^k \mid \boldsymbol{x}, \boldsymbol{w})}{\tau}\,, \tag{7}$$

where $sim(\cdot,\cdot)$ is a semantic similarity BERT-style model and τ is a temperature parameter. When output sequences $\boldsymbol{y}^n$ are sampled according to the posterior, the left term of Eq. (7) is equivalent to

Predictive Entropy. The right term of Eq. (7) can be interpreted as penalty that decreases uncertainty if there are many semantically similar answers. Therefore, SentenceSAR has a similar goal as Semantic Entropy, yet is more or less motivated heuristically.

The SAR method proposed in Duan et al. (2024) combines both SentenceSAR (Eq. (7)) and TokenSAR (Eq. (13)). We consider both SentenceSAR, TokenSAR and SAR in our experiments.

EigenScore. The EigenScore method proposed by Chen et al. (2024) operates in the latent space instead of output probabilities. Due to that, the method aims to better capture semantic information for an accurate assessment of an LLMs likelihood to hallucinate / confabulate. The EigenScore metric is defined as

$$\text{EigenScore} \;=\; \frac{1}{N}\log\det(\boldsymbol{\Sigma} + \alpha\, \boldsymbol{I}_N) \;=\; \frac{1}{N}\log(\prod_{k=1}^{N}\lambda_k) \;=\; \frac{1}{N}\sum_{k=1}^{N}\log(\lambda_k)\,, \tag{8}$$

where $\boldsymbol{\Sigma} = \boldsymbol{Z}^\top \cdot \boldsymbol{J}_d \cdot \boldsymbol{Z}$ is the covariance matrix, $\boldsymbol{Z}$ is a matrix of N sentence embeddings, taken from the latent space of the LLM with dimensionality d, $\boldsymbol{J}_d = \boldsymbol{I}_d - \frac{1}{d}\mathbf{1}_d$ is the centering matrix where $\boldsymbol{I}_d$ is an identity matrix and $\mathbf{1}_d$ is an all-one square matrix of size $d \times d$. The regularization term $\alpha \boldsymbol{I}_N$ with small constant α is added such that $\boldsymbol{\Sigma}$ has full rank. The set $\{\lambda_1, \lambda_2, ..., \Lambda_N\}$ denotes the eigenvalues of the regularized covariance matrix $\boldsymbol{\Sigma} + \alpha\, \boldsymbol{I}_N$, obtained through singular value decomposition. We followed the implementational details by the original authors Chen et al. (2024). Noteworthy, according to **Remark 1** in Chen et al. (2024), EigenScore is an approximation of differential entropy in the sequence embeddings space. It can thus be interpreted as a variant of Semantic Entropy, yet not computed in the output space, but in the embedding space. While Semantic Entropy and other methods operating in the output space hinge on the quality of the semantic similarity operation in the output space, EigenScore depends critically on the quality of the sentence embedding space of the LLM.

A.1.2 Single Output Sequence

In addition to measures of predictive uncertainty defined as expectations over the sequence probability, also other methods that only consider a single output sequence have been proposed.

Maximum Sequence Probability. Similar to the univariate classification setting, the Maximum Sequence Probability has been considered as a measure of uncertainty (Fadeeva et al., 2023). For numerical stability, the negative logarithm of the sequence probability is considered. Formally, the Maximum Sequence Probability (i.e. the negative logarithm thereof) is given by

$$\text{MSP} \;=\; -\max_{\boldsymbol{y}} \log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\,. \tag{9}$$

Recently, Aichberger et al. (2024a) has shown that Eq. (9) is a theoretically justified measure of uncertainty. Approximating Eq. (9) is similarly hard as for other measures of uncertainty in practice, as the autoregressive nature of LLMs makes it necessary to search for the most likely sequence. However, Aichberger et al. (2024a) show that the greedily decoded sequence leads to a very efficient estimate that performs very well in practice called `G-NLL`, which is defined as

$$\text{G-NLL} = -\sum_{t=1}^{T}\log\left(\max_{y_t} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})\right) \approx \text{MSP}\,. \tag{10}$$

Perplexity. Closely related, the perplexity of an output sequence has been considered as measure of uncertainty (Ren et al., 2023). Note that this is essentially length-normalization as given in Eq. (12), with opposite sign. The perplexity of a sequence $\boldsymbol{y}$ is given by

$$\text{PP} \;=\; \exp\left\{\frac{1}{T}\sum_{t=1}^{T} -\log p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})\right\}\,. \tag{11}$$

A.1.3 Weighting Token Probabilities

A fundamental problem of calculating uncertainty measures on a sequence basis instead of a token basis is, that there is a depenency on the sequence length T. Therefore, short answers are automatically less uncertain than long answers. An ad-hoc solution that is widely regarded in the literature

is to use length-normalization (see e.g. Cover & Thomas (2006)). Furthermore, alternatives to this indiscriminative normalization have been proposed, e.g. TokenSAR where individual tokens are weighted according to their semantic relevance (Duan et al., 2024).

Length-normalization. Malinin & Gales (2020) popularized the use of length-normalization to make Predictive Entropy comparable across sequence lengths. Instead of the usual sequence probability, the heuristic length-normalized probability distribution

$$\bar{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \;=\; \prod_{t=1}^{T} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})^{\frac{1}{T}} \;=\; \exp\left\{ \frac{1}{T} \sum_{t=1}^{T} \log p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w}) \right\} \tag{12}$$

is considered. Note that this distribution is therefore unnormalized in the sense that the sum over all sequences does not sum up to one. This heuristic has been widely used together with Predictive Entropy, Semantic Entropy or the Maximum Sequence Probability. Using $\bar{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ instead of $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ in their definitions essentially leads to an additional factor $\frac{1}{T}$ in the definitions of these uncertainty measures. Furthermore, we note that Perplexity is essentially the negative length-normalized sequence probability of an output sequence.

TokenSAR. In order to make uncertainty scores comparable across different sequence lengths, instead of summing up token log-likelihoods, one can calculate a weighted average. While length-normalization uniformly weights with one divided by the sequence length, the TokenSAR method by Duan et al. (2024) introduces a weighting dependent on input / output pair $\boldsymbol{x}, \boldsymbol{y}$. The TokenSAR score is given by

$$\text{TokenSAR} \;=\; \sum_{t=1}^{T} -\log p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w}) \; \frac{R(y_t, \boldsymbol{y}, \boldsymbol{x})}{\sum_{t=1}^{T} R(y_t, \boldsymbol{y}, \boldsymbol{x})} \tag{13}$$

with $R(y_t, \boldsymbol{y}, \boldsymbol{x}) \;=\; 1 - |sim(\boldsymbol{x} \circ \boldsymbol{y}, \boldsymbol{x} \circ \boldsymbol{y} \backslash \{y_t\})|$. The semantic similarity metric $sim(\cdot, \cdot)$ is a BERT-style model and $\circ$ denotes concatenating two token sequences. Essentially, the weighting term R captures the semantic similarity of an $\boldsymbol{x}, \boldsymbol{y}$ pair and itself, yet leaving out one token of the output sequence. If the similarity chances substantially when removing one token, this one is weighted higher in the weighted average.

The SAR method proposed in Duan et al. (2024) combines both SentenceSAR (Eq. (7)) and TokenSAR (Eq. (13)). We consider both SentenceSAR, TokenSAR and SAR in our experiments.

A.2 Heuristics

Furthermore, we consider popular heuristic methods, that are not grounded in information-theory.

p(True). Kadavath et al. (2022) introduced the p(True) baseline to assess the confidence of the model in its own response. The model first generates an answer to a question and then evaluates the probability p(True) — the likelihood that the answer is correct. This is done by prompting the model to assess its own output, such as asking whether the answer is "True" or "False", and using the probabilities assigned to these responses as a confidence score.

Length of generated answer. Another heuristic baseline is to consider the length of the generated answers. The reasoning behind this is, that if the model does not know an answer, it will generate longer and more meaningless content as is often observed in public debates. We are not aware of any prior work that has considered it as an uncertainty estimation heuristic, although sequence length plays a role in the analysis in (Santilli et al., 2024).

A.3 Notes on Correctness Functions Used in NLG

Artifacts in a widely used ROUGE-2 and BLEU Implementation. Notably, ROUGE-2 and BLEU show low agreement to other n-gram based metrics while showing some higher than average agreement to each other. Low agreement of BLEU to other metrics can in part be explained by correctness values being low, making the commonly used 0.5 threshold a poor choice. Upon closer inspection it turned out that the standard implementations of ROUGE and BLEU that is widely

Table 3: Accuracies of the models for evaluated datasets according to corresponding correctness functions. This table lists the dataset / model papers evaluated in this work. Nan values in SQUAD is expected behavior, as there are no correctness labels for the artificially unanswerable OOD part. Known-Unknown Amayuelas et al. (2024) dataset generations were performed without accuracy computation as it we used it as a strictly OOD detection dataset.

Dataset	Model / Temp	NaN Values	Accuracy	Correctness Function
BCB	Llama-3 8b / 1.	0	0.34	Exact
BCB	Llama-3 8b IT / 1.	2	0.21	Exact
COLLIE	Phi-3.5 / 1.	0	0.30	Exact
COLLIE	Llama-3 70b IT / 1.	0	0.49	Exact
COLLIE	Falcon Mamba / 1.	0	0.14	Exact
COLLIE	Llama-3 8b / 1.	0	0.145	Exact
COLLIE	Falcon Mamba IT / 1.	0	0.166	Exact
COLLIE	Llama-3 8b IT / 1.	0	0.42	Exact
COLLIE	Phi-3.5 IT / 1.	0	0.21	Exact
COQA	Llama-3 8b IT / 1.	0	0.86	MoJI
COQA	Phi-3.5 IT / 1.	0	0.81	MoJI
COQA	Llama-3 8b / 1.	0	0.54	MoJI
COQA	Llama-3 70b / 1.	0	0.73	MoJI
SQUAD	Phi-3.5 IT / 1.	5945	0.92	MoJI
SQUAD	Llama-3 8b IT / 1.	5945	0.94	MoJI
SQUAD	Llama-3 8b / 1.	5945	0.74	MoJI
SQUAD	Llama-3 70b IT / 1.	5945	0.94	MoJI
TRIVIA	Phi-3.5 IT / 1.	0	0.58	MoJI
TRIVIA	Llama-3 8b IT / 1.	0	0.74	MoJI

used in uncertainty estimation evaluation Luong et al. (2017) return correctness of zero if either the proposed or reference answers are shorter than a predefined n-gram, which is 2 for ROUGE-2 and 4 for BLEU. Considering the distribution of reference answers in QA datasets, this is a major artifact demanding attention.

B Details on Experimental Setting

To broaden the domain of the experiments, we have preferred smaller models from diverse families. We utilized the Llama-3 Dubey et al. (2024), Phi-3.5 Abdin et al. (2024) and Falcon Mamba Zuo et al. (2024) series of models. Falcon Mamba models, although less performant than their attention based counterparts, were utilized to broaden the evaluation coverage to the upcoming linear attention models. The dataset model pairs considered and the accuracies achieved on the most appropriate correctness metric are listed in Tab. 3.

B.1 Computing the Elo Rating

The Elo rating was computed as follows: the initial rating were initialized to 1000 for each method. For each step, a dataset / model pair was selected, as well as two distinct uncertainty estimation methods. Out of the two methods, one with higher AUC against the corresponding risk indicator would be considered the winner. The scores would then be updated according to the standard Elo update rule with $s = 400$ and $K = 2$. K value roughly corresponds to the update step size modifier. The relatively low value of K was selected since the optimization was performed for $100,000$ steps until convergence (Fig. 4). The mean and variance of the Elo scores over the last 1000 iterations were taken as the final values presented in Fig. 3.

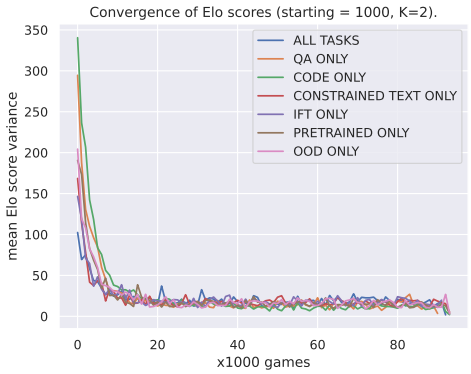

Figure 4: Convergence of Elo ratings on the various experimental subsets.

C Prompts Used for judge Models

All judge models were of Llama-3 family. All three sizes were considered: 8B, 70B, 405B. The length of the completion was observed to be largely in 2-3 token range, indicating that the prompting largely succeeded at imposing anticipated output structure onto the model.

Throughout our investigations, we use the Llama-3 8B and 70B Dubey et al. (2024), Phi-3.5 Abdin et al. (2024) and Falcon Mamba 7B Zuo et al. (2024) series of models, both pretrained and instruction tuned (IT). If the model is not specified explicitly, all models are considered. Further details such as the accuracies of individual models on the considered datasets are provided in Apx. B. We consider the QA datasets CoQA (Reddy et al., 2019), TriviaQA (Joshi et al., 2017) and SQuADv2 (Rajpurkar et al., 2018).

QA prompt follows the implementation of Farquhar et al. (2024):

```
We are assessing the quality of answers
to the following question: {question}
The expected answer is: {correct_answer}.
The proposed answer is: {predicted_answer}
Within the context of the question,
does the proposed answer mean the same as the expected answer?
Respond only with yes or no.
Response:
```

Gen prompt is derived from the QA prompt with minor modifications:

```
We are assessing the quality of answers
to the following question: {question}
The following are example answers: {correct_answer}.
The proposed answer is: {predicted_answer}
Within the context of the question and example answer,
is the proposed answer correct?
Respond only with yes or no.
Response:
```

D Comparison of LLM-as-a-judge to Each other and Exact Correctness

In Fig. 5 we investigate the consistency of correctness assessment between different judge models. We can observe, that even identical models can diverge based on the prompt. When the temperature is not set to 0, we are additionally facing variability due to sampling outputs from the judge model.

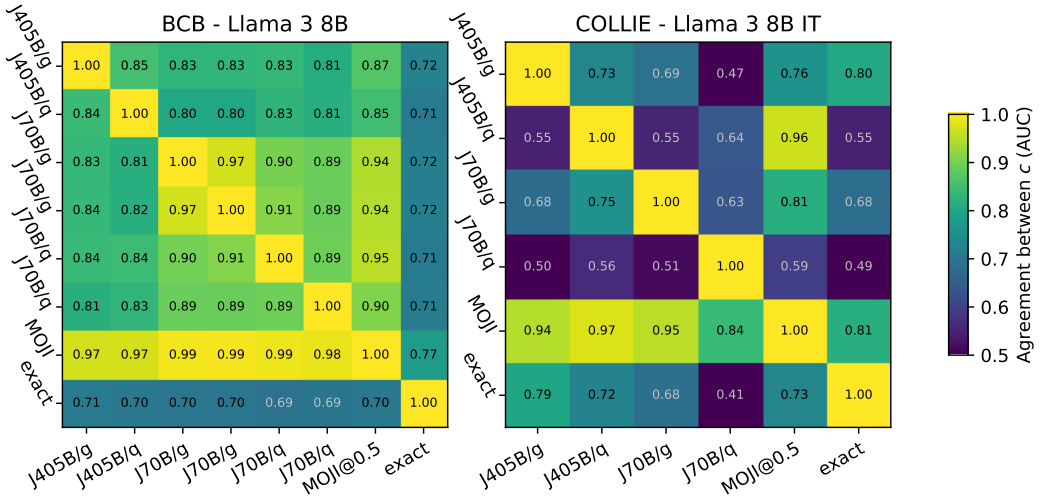

Figure 5: Agreement scores between judges on structured problems. The ticks indicate model size / prompt / sampling temperature used to assess correctness. judges of similar sizes tend to agree. Larger judge models tend to agree better with the exact solution, especially on COLLIE. The sampling temperature of the judge model appears to have a relatively minor effect on the outcome. Prompt affects the evaluation quality substantially, especially on COLLIE, which requires much less direct pattern matching and more reasoning.

E Additional Theoretical Considerations

E.1 Empirical Properties of Uncertainty Quantification Algorithms

According to the how uncertainty quantification algorithms are evaluated in the literature (Welling & Teh, 2011; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Malinin & Gales, 2018; D' Angelo & Fortuin, 2021; Daxberger et al., 2021; Mukhoti et al., 2023; Schweighofer et al., 2023), we can say that uncertainty is a function $\hat{u}_{\text{ale}}(\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}_u)$ (aleatoric) or $\hat{u}_{\text{epi}}(\boldsymbol{x}, \mathcal{D}; \boldsymbol{\theta}_u)$ (epistemic) with positive real valued codomain that has the following empirical properties:

1. $\hat{u}$ is higher for $\boldsymbol{x}' \sim \mathcal{D}_{\text{test}}$ than for $\boldsymbol{x} \sim \mathcal{D}_{\text{test}}$ if the risk of prediction using $\boldsymbol{w}$ (aleatoric) or $\boldsymbol{w} \sim p(\boldsymbol{w} \mid \mathcal{D})$ (epistemic) for $\boldsymbol{x}'$ is higher than for $\boldsymbol{x}$.
2. $\hat{u}$ is not lower for $\boldsymbol{x}'$ than for $\boldsymbol{x} \sim \mathcal{D}_{\text{test}}$ if $\boldsymbol{x}'$ is drawn from a different data generating function than one that produced the training data $\mathcal{D}$.
3. $\hat{u}$ is not lower for $\boldsymbol{x}'$ than for $\boldsymbol{x} \sim \mathcal{D}_{\text{test}}$ if $\boldsymbol{x}'$ is obtained from $\boldsymbol{x}$ by some perturbation.

These properties can be distilled from ubiquitously used evaluation protocols in uncertainty quantification literature in classification setting. Note that the first and third properties are characteristic of both aleatoric and epistemic uncertainty, whereas the second is usually attributed to epistemic uncertainty. In the classification setting, most of the literature is focused on epistemic uncertainty since it involves, depending on the definition, estimating a more difficult posterior integral of a divergence, which requires intricate posterior sampling techniques (Wilson & Izmailov, 2020; Schweighofer et al., 2023). The three empirical properties can be unified in terms of viewing uncertainty as an indicator of prediction risk (Kotelevskii & Panov, 2025; Lahlou et al., 2023).

Another assumption is sometimes used:

4. If $\hat{u}_{epi}$ is higher for $\boldsymbol{x}' \sim \mathcal{D}_{\text{domain}}$ than for $\boldsymbol{x}$, then adding the $(\boldsymbol{x}', \boldsymbol{y}')$ to the training dataset $\mathcal{D}$ would on expectation lead to higher risk reduction on $\mathcal{D}_{\text{domain}}$ than adding $(\boldsymbol{x}, \boldsymbol{y})$.

This is the active learning assumption which in classification literature is usually associated with the epistemic uncertainty (Kirsch, 2024). Active Learning evaluation is a challenging task with many

caveats even in the classification setting (Hacohen et al., 2022; Lüth et al., 2023). Furthermore it requires a true label and some degree of model tuning. Autoregressive generation further complicates this. Therefore we do not consider AL assumption for evaluation in our work. The way these assumptions are formulated implies that the correlation coefficient according to which they are evaluated must be invariant to monotone increasing transformations.

E.2 Effects of Noisy Reference on Rank Correlation

In this section we investigate the effects of the defects of the reference class labels on rank correlation. We specifically focus on AUC, as it is the rank correlation most commonly used in Uncertainty Estimation literature for risk correlation experiments. We show that both variance and bias in risk indicator values lead to biased AUC estimates. Both of the considered scenarios support using MoJI as the approximate correctness measure of choice.

E.2.1 Sample AUROC

Sample AUROC can be computed explicitly as follows:

$$\text{AUC}^{\text{s}} = \frac{1}{n_0 n_1} \sum_{i:y_i=1} \sum_{j:y_j=0} I(s_i > s_j) + 0.5 \cdot I(s_i = s_j) \tag{14}$$

It has an equivalent MC estimator that implies sampling positive-negative labeled pairs:

$$\text{AUC}^{\text{s-MC}} \approx \frac{1}{M} \sum_{i}^{M} I(s_i^1 > s_i^0) \tag{15}$$

The two forms are equivalent and are unbiased and consistent AUC estimators and are equivalent to the original rank based U statistic (Mann & Whitney, 1947). Generally, the AUC corresponds to the expected probability that the scorer $s : \mathcal{X} \to R$ ranks the items $(x_1, \dots x_n)$ in a way that those with positive binary labels $(y_1, \dots y_n)$ have higher score than ones with negative labels.

$$\text{AUC} = \text{E}_{x_p \sim p(x,y=0)} \text{E}_{x_n \sim p(x,y=1)} P\left[s(x_p) > s(x_n)\right] \tag{16}$$

In case of empirical assessment of the uncertainty estimation algorithm by correlation to risk (as per Sec.2 and Appx.E.1) ξ, the y labels are the negated correctness $\neg c$ and scores are the uncertainty estimates $\hat{u}$.

Sample AUROC with label noise Let us now consider scenario, where the reference labels are perturbed randomly by a Bernoulli noise:

$$c_{x_i}^{\text{noisy}} = \begin{cases} c_{x_i} & \text{if } \gamma \sim \mathcal{B}(p) = 0 \\ \neg c_{x_i} & \text{if } \gamma \sim \mathcal{B}(p) = 1 \end{cases} \tag{17}$$

Note, that rounded expectation of $c_{x_i}^{\text{noisy}}$ (its median) equals the true value of c_{x_i} if the noise magnitude $p < 0.5$:

$$\text{round}\left[\text{E}[c^{\text{noisy}}(x_i)]\right] = c(x_i) \tag{18}$$

Informally this can be viewed as an unbiased estimator of $c(x_i)$ with added variance for a binary variable. γ is independent of the example i to which it applies, contrary to the bias introduced by distortion in the previous section.

To inspect the properties of the AUC estimate in case of of noisy reference, we will use the AUC^{MC} from Eq.(15) formulation of the estimator, as the direct sample AUC estimation from Eq.(14) is less suitable for accommodating the noise term. In this regime we require sampling pairs of inputs with positive/negative label i. This assumes ability to specifically sample the positive or negative class, which we take for granted (i.e. class balance assumption) without additional importance sampling considerations. We decompose the Eq.(15) similarly to what we did for the bias case:

$$
\begin{aligned}
\text{AUC}^{\text{noisy-MC}} &= \\
&= \frac{1}{M}\sum_{i}^{M} I\left(s(x_i^a) > s_(x_i^b) \mid c^{\text{noisy}}(x_i^a) = 1, c^{\text{noisy}}(x_i^b) = 0\right) \\
&= \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 0\right) \cdot p(\gamma = 1)^2 + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p(\gamma = 0)p(\gamma = 1) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p(\gamma = 0)p(\gamma = 1) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 1\right) \cdot p(\gamma = 0)^2 \end{cases} \\
&= \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 0\right) \cdot (1-p)^2 + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 1\right) \cdot p^2 \end{cases}
\end{aligned} \tag{19}
$$

Note that the coefficients in sum up to 1, which makes sense. Then we can proceed by separating the part that corresponds to the AUC estimator with unbiased labels:

$$
\begin{aligned}
\text{AUC}^{\text{noisy-MC}} &= \\
&= \frac{1}{M}\sum_{i}^{M} I\left(s(x_i^a) > s_(x_i^b) \mid c^{\text{noisy}}(x_i^a) = 1, c^{\text{noisy}}(x_i^b) = 0\right) \\
&= \frac{1}{M}\sum_{i}^{M} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 0\right) \cdot (1-p)^2 + \\
&+ \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 1\right) \cdot p^2 \end{cases} \\
&= \text{AUC}^{\text{MC}} \cdot (1-p)^2 + \\
&+ \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 1\right) \cdot p^2 \end{cases} \\
&= \text{AUC}^{\text{MC}} \cdot (1-p)^2 + \\
&+ \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p \cdot (1-p) + \\ \left(1 - I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 0\right)\right) \cdot p^2 \end{cases} \\
&= \text{AUC}^{\text{MC}} \cdot (1-p)^2 + \frac{1}{M}\sum_{i}^{M} p^2 - \text{AUC}^{\text{MC}} \cdot p^2 + \\
&+ \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) \cdot p \cdot (1-p) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \cdot p \cdot (1-p) \end{cases} \\
&= \text{AUC}^{\text{MC}} \cdot (1-2p) + \frac{1}{M}\sum_{i}^{M} p^2 + \\
&+ p \cdot (1-p) \frac{1}{M}\sum_{i}^{M} \begin{cases} I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 1, c(x_i^b) = 1\right) + \\ I\left(s(x_i^a) > s_(x_i^b) \mid c(x_i^a) = 0, c(x_i^b) = 0\right) \end{cases}
\end{aligned} \tag{20}
$$

We can safely assume that the two identity terms within the classes sum up to 0.5 over large number of samples. This is because within the same class we can sample both (x_i^a, x_i^b) and (x_i^b, x_i^a) with the same likelihood.

$$
\begin{aligned}
\text{AUC}^{\text{noisy-MC}} &= \\
&= \text{AUC}^{\text{MC}} \cdot (1-2p) + \frac{1}{M}\sum_{i}^{M} p^2 + \sum_{i}^{M} \frac{1}{M} p \cdot (1-p) \\
&= \text{AUC}^{\text{MC}} \cdot (1-2p) + p
\end{aligned} \tag{21}
$$

While in Eq.(21) the first term is lower than the value obtained with unbiased labels by a factor of $1-2p$. With this, $\text{AUC}^{\text{noisy-MC}} = \text{AUC}^{\text{MC}}$ only when the $\text{AUC}^{\text{MC}} = 0.5$. Intuitively, we can see that random classifier will not be affected by noise in the labels. This shows, that ultimately, introducing random noise to the labels increases the bias of the AUC estimator and results in a loss of its asymptotic consistency. *In context of our work* this demonstrates that having variance in the risk indicator (i.e. stochastic approximate correctness) yields a biased estimate of ξ when using AUC as rank correlation. This is particularly relevant to samples from LLM-as-a-judge, which are rolled out stochastically.

Sample AUROC with biased labels Lets consider a scenario, where the correctness function is biased. This is equivalent to permanently perturbing the correctness labels $(c_1, \dots c_n)$ with some distortion function $d : X \mapsto \{0, 1\}$:

$$c^{\text{b}}_{x_i} = \begin{cases} c_{x_i} & \text{if } d_{x_i} = 0 \\ \neg c_{x_i} & \text{if } d_{x_i} = 1 \end{cases} \tag{22}$$

For brevity, we refer to $c^{\text{b}}_{x_i}$ as c_i and to d_{x_i} as d_i. Then (ignoring the ties for simplicity):

$$\begin{aligned}
\text{AUC}^{\text{s-b}} &= \\
&= \frac{1}{n_0 n_1} \sum_{i:y_i=1} \sum_{j:y_j=0} I(s_i > s_j) \\
&= \frac{1}{n_0} \sum_{i:y_i=1} \begin{cases} p(d_j = 0, y_j = 0) \sum_{j:y_j=0 \wedge d_j=0} I(s_i > s_j) + \\ p(d_j = 1, y_j = 1) \sum_{j:y_j=1 \wedge d_j=1} I(s_i > s_j) \end{cases} \\
&= \begin{cases} \sum_{i:y_i=1 \wedge d_i=0} \sum_{j:y_j=0 \wedge d_j=0} I(s_i > s_j) p(y_i = 1, d_i = 0) p(d_j = 0, y_j = 0) + \\ \sum_{i:y_i=0 \wedge d_i=1} \sum_{j:y_j=0 \wedge d_j=0} I(s_i > s_j) p(y_i = 0, d_i = 1) p(d_j = 0, y_j = 0) + \\ \sum_{i:y_i=1 \wedge d_i=0} \sum_{j:y_j=1 \wedge d_j=1} I(s_i > s_j) p(y_i = 1, d_i = 0) p(d_j = 1, y_j = 1) + \\ \sum_{i:y_i=0 \wedge d_i=1} \sum_{j:y_j=1 \wedge d_j=1} I(s_i > s_j) p(y_i = 0, d_i = 1) p(d_j = 1, y_j = 1) \end{cases} \\
&= \begin{cases} AUC^{\text{s}} \cdot p(d_i = 0, d_j = 0) + \\ 0.5 \cdot (p(d_i = 1, d_j = 0) + p(d_i = 0, d_j = 1)) + \\ (1 - AUC^{\text{s}}) \cdot p(d_i = 1, d_j = 1) \end{cases} \\
&= \begin{cases} AUC^{\text{s}} \cdot \frac{n(d_i=0) n(d_j=0)}{n_0 n_1} + \\ 0.5 \cdot \left(\frac{n(d_i=1) n(d_j=0)}{n_0 n_1} + \frac{n(d_i=0) n(d_j=1)}{n_0 n_1} \right) + \\ (1 - AUC^{\text{s}\,*}) \cdot \frac{n(d_i=1) n(d_j=1)}{n_0 n_1} \end{cases}
\end{aligned} \tag{23}$$

In the Eq.(23) we first decompose the AUC estimate with distorted labels into 4 terms. The first and the last term then can be expressed through the sample AUC with unbiased labels, which holds in the asymptotic case of large sample size ($N \to \inf$ where $N = n_0 + n_1$). The middle two terms equal 0.5 by symmetry argument.

$$\begin{aligned}
\text{AUC}^{\text{s-b}} &= \\
&= AUC^{\text{s}} \cdot \frac{n(d_i = 0) n(d_j = 0)}{n_0 n_1} + 0.5 \cdot \left(\frac{n(d_i = 1) n(d_j = 0)}{n_0 n_1} + \frac{n(d_i = 0) n(d_j = 1)}{n_0 n_1} \right) + \\
&+ (1 - AUC^{\text{s}\,*}) \cdot \frac{n(d_i = 1) n(d_j = 1)}{n_0 n_1} \\
&= AUC^{\text{s}} \cdot \frac{n(d_i = 0) n(d_j = 0)}{n_0 n_1} - AUC^{\text{s}\,*} \cdot \frac{n(d_i = 1) n(d_j = 1)}{n_0 n_1} + \\
&+ 0.5 \cdot \left(\frac{n(d_i = 1) n(d_j = 0)}{n_0 n_1} + \frac{n(d_i = 0) n(d_j = 1)}{n_0 n_1} \right) + \frac{n(d_i = 1) n(d_j = 1)}{n_0 n_1} \\
&= AUC^{\text{s}} \cdot \frac{n(d_i = 0) n(d_j = 0)}{n_0 n_1} - AUC^{\text{s}\,*} \cdot \frac{n(d_i = 1) n(d_j = 1)}{n_0 n_1} + \qquad (24) \\
&+ 0.5 \left(\frac{n(d_i = 1)}{n_0} + \frac{n(d_j = 1)}{n_1} \right) \qquad (25)
\end{aligned}$$

Here the $AUC^{\text{s}\,*}$ is the AUC of the subsample with flipped labels and AUC^{s} is the AUC of the undistorted part. Note, that in case of large sample size and random flipping of labels, this expression becomes equivalent to Eq.(21).

This shows, that the deviation from the original AUC depends on a) magnitude of distortion; b) on whether the AUC of distorted partition is similar to that of the undistorted partition. If the distortion is produced by random noise like in the previous section, the bias is higher if no resampling is done. This part of the identity above results in *In context of our work*, this shows that biased the risk indicator labels leads to bias and loss of consistency of the ξ estimate compared to the case of unbiased indicator.