Unconditional Truthfulness: Learning Conditional Dependency for Uncertainty Quantification of Large Language Models

Anonymous ACL submission

Abstract

Uncertainty quantification (UQ) is a perspective approach to detecting Large Language Model (LLM) hallucinations and low quality output. In this work, we address one of the challenges of UQ in generation tasks that arises from the conditional dependency between the generation steps of an LLM. We propose to learn this dependency from data. We train a regression model, which target variable is the gap between the conditional and the unconditional generation confidence. During LLM inference, we use this learned conditional dependency model to modulate the uncertainty of the current generation step based on the uncertainty of the previous step. Our experimental evaluation on nine datasets and three LLMs shows that the proposed method is highly effective for uncertainty quantification, achieving substantial improvements over rivaling approaches.

1 Introduction

Uncertainty quantification (UQ) (Gal and Ghahramani, 2016; Baan et al., 2023; Geng et al., 2023; Fadeeva et al., 2023) is of growing interest in the Natural Language Processing (NLP) community for dealing with Large Language Models (LLMs) hallucinations (Fadeeva et al., 2024) and low quality generations (Malinin and Gales, 2021) in an efficient manner. For example, high uncertainty could serve as an indicator that the entire generation should be discarded as potentially harmful to users (selective generation), or that a part of the generation should be flagged as untrustworthy.

There are many approaches for detecting hallucinations and low-quality outputs of LLMs (Manakul et al., 2023; Min et al., 2023; Chen et al., 2023). However, the majority of them leverage external knowledge sources or a second LLM. Knowledge sources are generally patchy in coverage while censoring the outputs of a small LLM using a bigger one has a high computational cost and is impractical. We argue that models inherently contain information about their own knowledge limitations, and that there should be an efficient way to access this information, which can enable LLM-based applications that are both safe and practical.

For general classification and regression tasks and for text classification in particular, there is a well-developed battery of UQ techniques (Zhang et al., 2019; He et al., 2020; Xin et al., 2021; Wang et al., 2022; Vazhentsev et al., 2023; He et al., 2024). For text generation tasks, UQ is much more complicated. The complexity is multifold: (1) there is an infinite number of possible generations, which complicates the normalization of the uncertainty scores; (2) in the general case, there are an infinite number of correct answers; (3) decisions are generally based on imprecise sampling and inference algorithms such as beam search; (4) there is not one, but multiple tokens, and the uncertainty of these predictions need to be aggregated; and (5) finally, the predictions at each generation step are not conditionally independent (Zhang et al., 2023).

This last problem is the focus of the present work. During generation, LLMs condition on the previously-generated tokens. Thus, if an LLM has hallucinated and generated an incorrect claim at the beginning or middle of the sequence, all subsequently generated claims might also be incorrect. Even in the case when the first claim was generated with high uncertainty, this is not taken into account during the subsequent generation process. This means that while the first error could be implicitly recognized as such with high uncertainty, all subsequent mistakes will be overlooked, because the generation process conditioned on this error will be very confident.

Below, we suggest a theoretically-motivated data-driven solution to this problem. We note that the attention between generated tokens provides information about the conditional dependency between the generation steps. Previously,

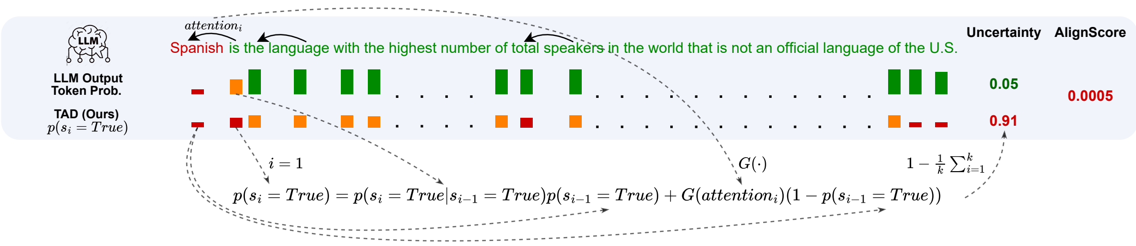

Figure 1: An illustration of the proposed method TAD. The figure depicts generated tokens, uncertainty scores for the generated sequence, and probabilities assigned by an LLM and TAD (represented with bars). The output was generated by Gemma 7b for the question *What is the language with the highest number of total speakers in the world that is not an official language of the U.S.?* The LLM starts with generating a token *Spanish* that leads to the erroneous answer. The probabilities estimated by the LLM are high for all tokens except for the first one, which makes the uncertainty scores based on raw probabilities misleadingly low. On the contrary, TAD takes into account uncertainty from the previous step using a trainable model $G(\cdot)$ based on attention, resulting in a high overall uncertainty for the generated answer.

there have been several attempts to suggest heuristic approaches to model this dependency (Zhang et al., 2023). We argue that the particular algorithmic function would be too difficult to engineer, and thus we propose to learn this dependency from data. For this purpose, we generate a training dataset with a target variable representing the gap between the conditional generation confidence and the unconditional confidence. The latter refers to the probability that a statement is correct without considering any context or previously generated statements, which may be inaccurate or erroneous. Using attention-based features, we trained an ML-based regression model to predict this gap that is further used for modifying the certainty of the current generation. We use attention-based features to ensure the generalizability of such an approach, supporting the training of a robust conditional dependency model. We call the proposed approach *trainable attention-based dependency (TAD)*. Figure 1 illustrates the idea behind the proposed method on the real output of an LLM. Our extensive experiments demonstrate that TAD offers substantial improvements in UQ over the baselines in tasks where an LLM is required to generate long sequences.

The contributions of this work are as follows:

- A new data-driven approach to uncertainty quantification that models the conditional dependency between the individual token predictions of an LLM.
- A computationally-efficient implementation of the method that leverages simple linear regression, making it practical for real-world applications based on LLMs.
- An empirical demonstration that the proposed method outperforms previous approaches across nine datasets and three LLMs.

2 Related Work

With the advent of LLMs, UQ has become an urgent research problem in NLP. As previously mentioned, this area not only offers promising practical benefits, but it also presents several intriguing research challenges. The majority of methods for UQ of LLM generations has been unsupervised, with few recently-proposed supervised methods.

Unsupervised UQ methods. Several methods adapt information-based UQ techniques by aggregating logits of generated tokens in various ways. Fomicheva et al. (2020) experimented with perplexity and mean token entropy for MT quality estimation. Takayama and Arase (2019) adapted point-wise mutual information (PMI), and van der Poel et al. (2022) extended this approach to conditional PMI. The advantages of these techniques are their simplicity, usually minimal computational overhead, and robust performance. A well-known approach to UQ in general is ensembling (Lakshminarayanan et al., 2017) and Monte Carlo (MC) dropout (Gal and Ghahramani, 2016). Malinin and Gales (2021) and Fomicheva et al. (2020) adapted it to sequence generation problems. In this category, lexical similarity (Fomicheva et al., 2020) is a very competitive baseline that can be applied to black-box models (without any access to logits or internal model representations).

The problem of multiple correct generations was explicitly addressed in (Kuhn et al., 2023; Nikitin et al., 2024; Cheng and Vlachos, 2024) and in a series of black-box generation methods (Lin et al., 2023). The main idea is to sample multiple generations from an LLM, extract semantically equivalent clusters, and analyze the diversity of the generated meanings instead of the surface forms.

Fadeeva et al. (2024) addressed the problem of multiple sources of uncertainty present in the LLM probability distribution that are irrelevant for hallucination detection and fact-checking. In addition to dealing with multiple correct generations, they also suggested mitigating the influence of the uncertainty related to the type of generated claims.

Zhang et al. (2023) and Duan et al. (2023) emphasized that not all generated tokens should contribute to the uncertainty score for the entire generated text and proposed various heuristics to select only relevant tokens. Zhang et al. (2023) also modeled the conditional dependency between the generation steps by adding a penalty to an uncertainty score that depends on the uncertainties of previously-generated tokens. The penalty depends on max-pooled attention to previous tokens from the current generation step.

Overall, most previous work on UQ has not addressed the conditional dependency between the predictions, or has addressed it using heuristics. We argue that the conditional dependency is an important aspect of UQ for text generation tasks and we propose a data-driven approach to it. We also note that techniques based on sampling multiple answers from LLMs usually introduce prohibitive computational overhead. We argue that for UQ methods to be practical, they should also be computationally efficient.

Supervised UQ methods. Supervised regression-based confidence estimators are well-known for classification problems, primarily from computer vision (Lahlou et al., 2022; Park and Blei, 2024). Their key benefit is computational efficiency.

A handful of papers applied this approach to text generation tasks. Lu et al. (2022) proposed to train a regression head of a model to predict confidence. They noted that the probability distribution of a language model is poorly calibrated and cannot be used directly to spot low quality translations. They trained an additional head by modifying the loss function and adding a regularizer. However, their approach is only applicable when fine-tuning language models for Machine Translation (MT), and is not suitable for general-purpose instruction-tuned LLMs. In a similar vein, Azaria and Mitchell (2023) approached the task of UQ by training a multi-layer perceptron (MLP) on the activations of the internal layers of LLMs. For this purpose, they annotated a dataset of true and false statements, and used forced LM decoding to obtain model outputs. They evaluated the ability of the trained MLP to classify the statements as true or false and demonstrated that it outperforms other supervised baselines and few-shot prompting of the LLM itself. However, due to the reliance on forced decoding, their experimental setup is far from real-world hallucination detection, where an LLM can perform unrestricted generation. Another limitation is that their method can provide veracity scores only for the entire generated text.

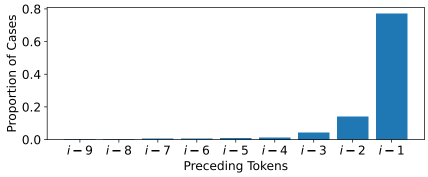

Figure 2: The fraction of cases where Gemma 7b pays the greatest attention on the corresponding previous token when generates a token t_i. We use attention weight matrices from all layers after max-pooling across attention heads. The test instances are from the TruthfulQA dataset (Lin et al., 2022).

Unlike these methods, besides learning uncertainty scores directly from data, we also learn the conditional dependency between the generation steps. Our method is also flexible as it can be used on various levels: for the entire text, at the sub-sentence level, or for individual tokens.

3 Trainable Attention-Based Conditional Dependency

In this section, we present our approach to learn the conditional dependency between the generation steps and our UQ method based on it.

3.1 Theoretical Background and Motivation

When an LLM generates a sequence of tokens t_i, it provides us a conditional probability distribution $p(t_i \mid t_{<i})$. This essentially means the LLM considers that everything generated so far is correct, which might not be the case. In practice, we would like to somehow propagate its uncertainty from previous generation steps.

In general, the probability distribution $p(t_i \mid t_{<i})$ is conditioned on all previously generated tokens $t_{<i}$. Nevertheless, empirical evidence shows that

in the majority of cases, LLMs pay the most attention to the last preceding token. Figure 2 illustrates that for 76% of cases, the greatest attention is paid towards the previous token, while for other tokens, the attention is significantly lower. For the sake of simplicity, we assume that only the uncertainty from the previous tokens is propagated to the current generation step. This assumption leads us to the first-order Markov process, in which the probability for the token t_i is conditioned only on the token t_{i-1}. This assumption can be expressed as follows: $p(t_i \mid t_{<i}) \simeq p(t_i \mid t_{i-1})$.

For simplicity, consider that we have trained an LLM that generates only tokens true ('T') or false ('F'). The probability of the token t_i being 'T' is given by the conditional probability $p(t_i \mid t_{i-1}) = p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})$. Assume we already have some tokens $t_1, t_2, \ldots, t_n$ and a prompt x. At each step, based on Markov process assumptions, the LLM provides us $p(t_1 = \mathrm{T} \mid x), p(t_2 = \mathrm{T} \mid t_1 = \mathrm{T}), \ldots, p(t_n = \mathrm{T} \mid t_{n-1} = \mathrm{T})$.

These probability distributions are conditionally dependent on the previous ones. However, to estimate the correctness of some token t_i, we need to obtain an *unconditional probability* $p(t_i = \mathrm{T})$. The LLM does not provide such probability during the standard generation process. Some heuristic techniques such as P(true) (Kadavath et al., 2022) can estimate the unconditional probability through rerunning an LLM on the generated text. However, it introduces expensive overhead, which approximately doubles the generation time and is not appropriate for token-level estimation. We would like to have a computationally efficient approach that does not need rerunning the LLM. Let us expand $p(t_i = \mathrm{T})$ according to the formula of full probability and express it using conditional probability:

$$\begin{aligned}
&p(t_i = \mathrm{T}) \\
&= p(t_i = \mathrm{T}, t_{i-1} = \mathrm{T}) + p(t_i = \mathrm{T}, t_{i-1} = \mathrm{F}) \\
&= p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})\ p(t_{i-1} = \mathrm{T}) + \\
&+ p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F})\ p(t_{i-1} = \mathrm{F}) \\
&= p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})\ p(t_{i-1} = \mathrm{T}) + \\
&+ p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F})\big(1 - p(t_{i-1} = \mathrm{T})\big). \quad (1)
\end{aligned}$$

In the obtained formula, $p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})$ is what the LLM provides during the current generation step in accordance with the specified assumptions. Consider that we know $p(t_{i-1} = \mathrm{T})$ as it is calculated on the previous generation step. We still do not know the remaining term: $p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F})$. Let us express it from the equation:

$$\begin{aligned}
&p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F}) \quad (2) \\
&= \frac{p(t_i = \mathrm{T}) - p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})\, p(t_{i-1} = \mathrm{T})}{1 - p(t_{i-1} = \mathrm{T})}.
\end{aligned}$$

This expression still requires $p(t_i = \mathrm{T})$, which is not known during the inference. However, we can replace it with some surrogate and use this expression to approximate $p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F})$ with a trainable model $G(Atten_i, p(t_{i-1} = \mathrm{T}), p(t_i = \mathrm{T} \mid s_{i-1} = \mathrm{T}))$. This function in fact measures the conditional dependency of the current generation step i on the previous one $i-1$. For model features, we suggest using attention from the step i to $i-1$: $Atten_i$, which is a vector of values taken from the attention matrices. We use the attention weight after the softmax from the previous token t_{i-1} to the current token t_i from all the layers and attention heads. The training data for this model could be obtained using equation (2) in the "offline" mode, where we do not care about efficiency of obtaining $p(t_i = \mathrm{T})$. We also note that if the implementation of G is a linear regression or a small neural network, it will not introduce much overhead to compute during the inference of the main LLM.

Finally, to obtain the confidence estimate, we replace $p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{F})$ with G in equation (1):

$$\begin{aligned}
&p(t_i = \mathrm{T}) = p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})\, p(t_{i-1} = \mathrm{T}) \\
&+ G\big(Atten_i, p(t_{i-1} = \mathrm{T}), p(t_i = \mathrm{T} \mid t_{i-1} = \mathrm{T})\big) \\
&\cdot \big(1 - p(t_{i-1} = \mathrm{T})\big). \quad (3)
\end{aligned}$$

3.2 Implementation

We implement the proposed method for token-level UQ and aggregate token-level scores into a score for the whole sequence.

Obtaining unconditional probability. To obtain the surrogate for the unconditional probability $\hat{p}(t_i)$ for a generated token t_i during the training phase, we use two strategies. The first one relies solely on the strict criterion of the presence of an existing token t_i in the ground truth text y:

$$\hat{p}(t_i) = \begin{cases} 1, t_i \in y, \\ 0, \text{otherwise.} \end{cases} \quad (4)$$

The second strategy additionally leverages AlignScore (Zha et al., 2023) $sim(\tilde{y}, y)$ between the generated text $\tilde{y}$ and the ground-truth y:

$$\hat{p}(t_i) = \begin{cases} \frac{1+sim(\tilde{y},y)}{2}, t_i \in y, \\ sim(\tilde{y}, y), \text{otherwise.} \end{cases} \quad (5)$$

This strategy aims to correct the target when a generated token is not present in the expected text, but the AlignScore is high, indicating that the generated text has the similar meaning as the training sentence. In the inverse situation, when the token is present, but the whole generation according to AlignScore is wrong, it penalizes the target.

Generating training data for TAD. We generate the training data for TAD using the original textual training dataset in the following way:

1. For the input prompt x_k and the target text y_k, using an LLM, we generate a text $\tilde{y}_k = t_1 t_2 \dots t_{n_k}$ of some length n_k and token probabilities $p(t_i \mid t_{<i})$.
2. For the first generated token t_1 in each text, we define its unconditional probability as a ground truth surrogate $p(t_1) = \hat{p}(t_1)$ according to formulas (4) or (5).
3. For each generated token t_i, $i = 2, \dots, n_k$:
   (a) We obtain $p(t_{i-1})$ from the previous generation step.
   (b) We define its unconditional probability as a ground truth surrogate $p(t_i) = \hat{p}(t_i)$ according to equations (4) or (5).
   (c) We compute the target variable for the function G using equation (2): $\tilde{G}_i = \frac{p(t_i) - p(t_i|t_{<i})p(t_{i-1})}{1 - p(t_{i-1})}$.

As a result, for each instance in the training dataset, we generate a sequence of target variables $\tilde{G}_i^k$ $k = 1, \dots, K, i = 1, \dots, n_k$. We further train the model G on these targets.

Model for G and its training procedure. We experiment with several regression models for TAD: liner regression (LinReg), CatBoost regression (Prokhorenkova et al., 2018), and a multi-layer perceptron (MLP). The hyperparameters of the regressors are obtained using cross-validation with five folds on the training dataset. We select the optimal values of the hyperparameters based on the best average PRR-AlignScore. Finally, we use these values to train the regression model on the full training set. The selected hyperparameters for the TAD modules are presented in Appendix C.1.

Inference procedure. During inference, we obtain predictions from the LLM as always, but we also extract features from the attention outputs. The features are used to compute G and a confidence score based on Equation (3).

4 Experiments

4.1 Experimental Setup

For experimental evaluation, we use the LM-Polygraph framework (Fadeeva et al., 2023). We focus on the task of selective generation (Ren et al., 2023) where we “reject” generated sequences due to low quality based on uncertainty scores. Rejecting means that we do not use the model output, and the corresponding queries are processed differently: e.g., they could be further reprocessed manually.

Metrics. Following previous work on UQ in text generation (Malinin and Gales, 2021; Fadeeva et al., 2023), we compare UQ methods using the Prediction Rejection Ratio (PRR) metric. PRR quantifies how well an uncertainty score can identify and reject low-quality predictions according to some quality metric. The PRR scores are normalized to the range $[0, 1]$ by linearly scaling the area under the PR curve between the values obtained with random selection (corresponding to 0) and oracle selection (corresponding to 1). Higher PRR values indicate better quality of selective generation. We use ROUGE-L, Accuracy, and AlignScore (Zha et al., 2023) as generation quality metrics.

Datasets. We consider three text generation tasks: text summarization (TS), QA with long free-form answers, and QA with free-form short answers, and for each task, we consider three datasets. Statistics about the datasets are provided in Table 18 in Appendix D. For TS, we experiment with CNN/DailyMail (See et al., 2017), XSum (Narayan et al., 2018), and SamSum (Gliwa et al., 2019). For the long answer QA task, we use PubMedQA (Jin et al., 2019), a QA dataset in the biomedical domain, with the task to answer biomedical research questions using the corresponding abstracts. We further use MedQUAD (Abacha and Demner-Fushman, 2019), which consists of real medical questions, and TruthfulQA (Lin et al., 2022), which consists of questions that some people would answer incorrectly due to a false belief or a misconception. For the QA task with short answers, we follow previous work on UQ (Kuhn et al., 2023; Duan et al., 2023; Lin et al., 2023) and we use three datasets: SciQ (Welbl et al., 2017), CoQA (Reddy et al., 2019), and TriviaQA (Joshi et al., 2017).

LLMs. We experiment with three LLMs: Gemma 7b (Mesnard et al., 2024), LLaMA 8b v3, and StableLM 12b v2 (Bellagente et al., 2024). The

UQ Method	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
MSP	-.329	-.116	.234	.177	-.039	.043	-.455	-.154	-.454	.008	.520	.268	**.699**	**.626**	.806	.744	**.828**	**.805**	8.61
Perplexity	-.358	-.179	.206	.291	.071	-.012	.527	.159	**.801**	.346	.381	.318	.458	.439	-.321	-.399	.820	.791	7.78
Mean Token Entropy	-.350	-.181	.172	.281	.082	-.017	.524	.147	.776	.330	.228	.290	.327	.339	-.368	-.398	.806	.786	8.94
Focus	-.324	-.161	.169	.232	.023	.008	-.357	-.146	-.408	-.100	.306	.298	.322	.250	-.098	.070	.651	.702	13.00
NumSemSets	.054	.049	.176	.176	.029	.052	.041	.017	-.067	.047	.132	.231	.203	.349	.132	.275	.677	.714	10.72
DegMat	.025	.060	.141	.161	.072	.088	.028	.008	-.063	.087	.211	.285	.345	.496	.401	.553	.740	.770	8.61
Eccentricity	-.055	.010	.059	.052	.028	-.005	-.016	-.011	-.144	.027	.116	.213	.514	.559	.487	.570	.737	.739	11.11
EigValLaplacian	.024	.063	.140	.156	.071	.087	.016	.004	-.155	.064	.200	.279	.479	.538	.507	.603	.727	.760	9.00
Lexical Similarity	.076	-.024	.256	.233	.108	.066	.068	.023	.240	-.024	.145	.117	.504	.499	.488	.538	.730	.734	8.78
MC NSE	-.005	-.023	.212	.195	.108	.102	.074	.012	-.000	.011	.076	.221	.440	.432	.357	.398	.727	.715	10.00
MC SE	.035	-.001	.251	.195	.123	.086	-.014	-.007	-.099	.013	.160	.141	.553	.514	.542	.557	.723	.712	9.11
Semantic Entropy	.034	.001	.250	.195	.110	.082	-.019	-.003	-.097	.019	.158	.159	.583	.566	.589	.605	.752	.745	8.28
SentenceSAR	-.077	-.037	.168	.133	.061	.090	-.072	-.033	-.221	.013	.305	.199	.643	.605	.700	.692	.792	.786	9.06
SAR	.042	-.006	.248	.245	.123	**.103**	.111	.014	.066	.035	.155	.263	.477	.503	.453	.515	.769	.770	7.11
TAD (LinReg)	.502	.257	.329	.263	**.177**	.078	**.576**	**.242**	.787	.376	**.563**	.294	.671	.608	.820	.751	.782	.760	3.00
TAD (LinReg+AlignScore)	**.541**	**.380**	**.353**	**.349**	.146	.092	.007	.064	.491	**.472**	.505	**.368**	.671	.600	**.834**	**.777**	.784	.766	**2.89**

Table 1: PRR↑ of UQ methods for the Gemma 7b model. Warmer colors indicate better results. The best method is in bold, the second best is underlined.

inference hyperparameters of the LLMs are given in Table 17 in Appendix C.2.

UQ baselines. We compare TAD to Maximum Sequence Probability (MSP), Mean Token Entropy, and Perplexity (Fomicheva et al., 2020), which are considered simple yet strong and robust baselines for selective generation across various tasks (Fadeeva et al., 2023). We also compare our method to more complex techniques, considered to be state-of-the-art UQ methods for LLMs: Lexical Similarity based on ROUGE-L (Fomicheva et al., 2020), Monte Carlo Sequence Entropy (MC SE), Monte Carlo Normalized Sequence Entropy (MC NSE; Kuhn et al. (2023)), black-box methods (NumSemSets, DegMat, Eccentricity, EigValLaplacian; Lin et al. (2023)), Semantic Entropy (Kuhn et al., 2023), hallucination detection with stronger focus (Focus; Zhang et al. (2023)), and Shifting Attention to Relevance (SAR; Duan et al. (2023)). For these methods, we generate five samples.

4.2 Main Results

Fine-grained comparison with the baselines. Tables 1, 7 and 8 in Appendix A present the results for Gemma 7b, Llama 8b v3, and StableLM 12b v2 models respectively.

We can see that for all summarization datasets, in the majority of cases, TAD outperforms the state-of-the-art methods by a large margin in terms of both considered metrics. The only exception is the case of PRR-AlignScore for StableLM on the XSum dataset, where SAR and Lexical Similarity are marginally better. At the same time, TAD confidently outperforms them in terms of PRR-ROUGE-L. In experiments with two other models on XSum, TAD also demonstrates large improvements in terms of both metrics over the baselines, which typically perform no better than a random choice. For example, TAD LinReg+AlignScore outperforms the second best baseline by .317 PRR-AlignScore and by .465 PRR-ROUGE-L absolute.

For QA with long answer datasets (PubMedQA, MedQUAD, and TruthfulQA), we see that TAD also confidently outperforms the baselines for all considered settings except for the experiment on TruthfulQA with LLaMA 8b v3 and for PRR-ROUGE-L measured on MedQUAD for Gemma. For example, in the experiment with LLaMA 8b v3 on PubMedQA, TAD outperforms the second best baseline – Perplexity by .190 of PRR-ROUGE-L and by .187 of PRR-AlignScore. For StableLM, the improvement is .049 of PRR-ROUGE-L and .083 of PRR-AlignScore. Additionally, we can see that on this task, the majority of sophisticated UQ baselines consistently fall behind simple techniques.

Finally, for QA with short answers (CoQA, SciQ, and TriviaQA), we can see that TAD notably outperforms baselines for all considered LLMs only on the SciQ dataset. TAD also marginally outperforms baselines in the experiments on CoQA with StableLM and Llama 8b v3. The lower performance on tasks with short answers is expected, since TAD primarily aims at improving the performance for tasks with long generations and complex conditional dependencies. Moreover, we can see that in the short-answer setting on TriviaQA and CoQA, the simplest baseline MSP demonstrates very strong performance, which is often the best.

When comparing the two strategies for obtaining the unconditional probability during training, we see that adding AlignScore usually helps for summarization, but it has a negative impact for QA.

Overall results. Table 2 presents the mean rank of each method aggregated over all datasets for each model separately. The lower rank is better. The column "Mean Rank" corresponds to the mean rank of the ranks across all models. Figure 3 additionally summarizes all experimental setups. Each cell presents a win rate for a method from a row compared to a method from a column. The aggre-

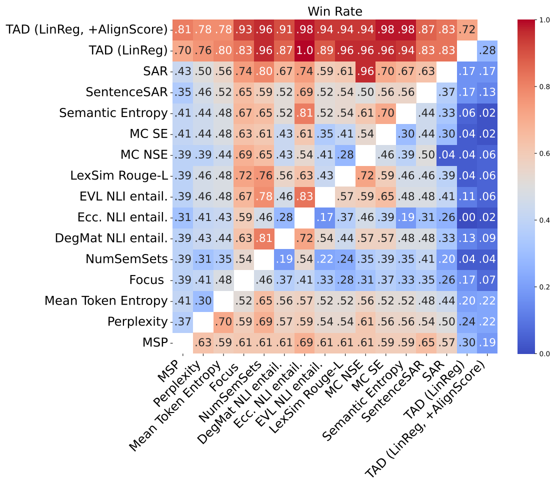

Figure 3: Summary of 54 experimental setups with various models and datasets. Each cell in the diagram presents the fraction of experiments where a method from a row outperforms a method from a column. Warmer colors indicate better results.

UQ Method	Gemma 7b	Llama-3 8b	StableLM 12b	Mean Rank
MSP	8.61	7.17	6.83	4.50
Perplexity	7.78	8.44	8.33	5.33
Mean Token Entropy	8.94	9.11	9.00	9.00
Focus	13.00	9.50	10.50	13.67
NumSemSets	10.72	10.78	12.83	15.00
DegMat	8.61	8.83	9.33	8.17
Eccentricity	11.11	11.33	11.61	15.33
EigValLaplacian	9.00	7.94	8.78	7.67
Lexical Similarity	8.78	9.22	8.56	8.33
MC NSE	10.00	10.72	10.22	13.00
MC SE	9.11	10.22	10.67	13.00
Semantic Entropy	8.28	9.06	9.06	7.67
SentenceSAR	9.06	9.39	8.22	9.00
SAR	7.11	7.78	6.33	3.33
TAD (LinReg)	3.00	3.72	3.50	2.00
TAD (LinReg+AlignSc.)	**2.89**	**2.78**	**2.22**	**1.00**

Table 2: Mean ranks of UQ methods aggregated over all datasets for each LLM separately (the lower the better). The column "Mean Rank" corresponds to the mean rank of the ranks across all LLMs. The best method is in bold, the second best is underlined.

gated results emphasize the significance of the performance improvements of the proposed method. Despite some baselines might show good results in individual cases, they usually are quite unstable resulting in poor overall ranking. TAD demonstrates more robust improvements across multiple tasks and LLMs, making it a better choice overall.

Generalization of TAD on unseen datasets. Tables 3, 9 and 10 in Appendix A.2 compare the results of TAD trained on a single in-domain training dataset to the results of TAD trained on all training datasets except one that represents the in-domain dataset for testing (we designate it as Gen TAD). This setting evaluates the out-of-domain performance of TAD. TAD without the AlignScore target demonstrates good generalization for QA with

UQ Method	XSUM		PubMedQA		CoQA		Mean
	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	Acc.	AlignSc.	Rank
MSP	-.356	-.153	-.024	.033	.648	.557	5.33
Focus	-.356	-.110	.045	-.063	.336	.261	6.50
SAR	-.029	.038	.075	.012	.474	.489	5.17
TAD (LinReg)	.358	.223	**.429**	**.220**	.639	.561	**2.17**
TAD (LinReg+AlignSc.)	**.579**	**.345**	-.018	.083	.657	**.567**	2.67
Gen. TAD (LinReg)	.006	-.032	.256	.208	.672	.541	3.33
Gen. TAD (LinReg+AlignSc.)	.210	.108	.179	.096	**.675**	.547	2.83

Table 3: The comparison of TAD trained on in-domain data with TAD trained on all out-of-domain datasets (designated with "Gen.") (PRR↑, Llama 8b v3). The best method is in bold, the second best is underlined.

long answers. Despite the results degrade on the unseen dataset, TAD confidently outperforms other baselines. Adding AlignScore for QA worsens the results probably due to overfitting.

For the TS task, on the contrary, adding AlignScore helps to achieve some generalization. The results substantially degrade, but are still better than for other baselines. On the short-answer QA task, training on out-of-domain data slightly improves PRR-Accuracy. More details about these experiments are presented in Appendix A.2.

4.3 Ablation Studies

Regression models and aggregation approaches. Detailed results with various regression models and aggregation approaches are presented in Table 4 and in Tables 11 and 12 in Appendix A. The optimal values of the hyper-parameters of TAD for all experimental setups are presented in Tables 14 to 16 in Appendix C.1 for Gemma 7b, LLaMA 8b v3, and StableLM 12b v2 models, respectively.

The results show that TAD based on regression using MLP and LinReg consistently outperform TAD based on CatBoost (Prokhorenkova et al., 2018). However, there is no big difference between MLP and LinReg. Therefore, for simplicity, we use LinReg as a regression method for TAD.

We investigate two strategies for aggregation of token-level TAD scores: the mean of the scores and the sum of the log scores inspired by perplexity. For the majority of the considered settings, the mean of the probabilities yields the best results. However, for QA with short answers, the sum of the log probabilities performs slightly better.

Comparison of features. Table 5 presents the experiments with various features for the regression model. For "TAD Embeds.", we utilize the embeddings from the last hidden state from the decoder. For "TAD Probs.", we use only generated probabilities for current and previous tokens, and $p(s_{i-1} = \mathrm{T})$. For "TAD Attn. Only", we use at-

UQ Method	Aggregation	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
		ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.496	.215	.201	.248	.064	-.011	.540	.181	.792	.382	.414	.283	.632	.578	.687	.634	.816	.800	5.89
TAD (CatBoost+AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.332	.146	.211	.269	.052	-.012	.556	.215	.665	.357	.382	.310	.603	.550	.550	.529	**.818**	**.801**	6.67
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	.324	.284	.100	.075	-.078	**.107**	-.373	-.112	-.461	.011	.452	.163	.669	.609	.810	.736	.792	.776	7.33
TAD (CatBoost+AlignScore)	$\sum_{k=1}^{K} \log p_k$	.249	.297	.057	.039	-.169	.093	-.573	-.190	-.472	-.001	.310	.083	**.717**	.626	.830	.774	.789	.775	8.28
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.502	.257	.329	.263	.177	.078	**.576**	**.242**	.787	.376	.563	.294	.510	.488	.619	.585	.811	.789	5.39
TAD (LinReg+AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	**.541**	**.380**	**.353**	**.349**	.146	.092	.007	.064	.491	**.472**	.505	**.368**	.471	.441	.484	.462	.805	.782	5.17
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	.396	.319	.072	.090	-.029	.092	-.387	-.116	-.460	.012	.573	.224	.671	.608	.820	.751	.782	.760	7.22
TAD (LinReg+AlignScore)	$\sum_{k=1}^{K} \log p_k$	.373	.351	.176	.121	-.099	.101	-.569	-.198	-.473	.000	.430	.187	.671	.600	.834	.777	.784	.766	7.22
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.504	.249	.246	.210	**.180**	.080	.564	.217	**.794**	.369	**.577**	.298	.665	.605	.686	.641	.813	.794	**4.50**
TAD (MLP+AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.536	.349	.321	.327	.118	.092	-.059	.021	.624	.418	.419	.298	.614	.559	.608	.590	.804	.781	5.56
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	.380	.301	.052	.042	-.020	.090	-.359	-.112	-.461	.010	.509	.183	.675	.613	.821	.754	.787	.764	7.28
TAD (MLP+AlignScore)	$\sum_{k=1}^{K} \log p_k$	.363	.340	.162	.105	-.100	.099	-.567	-.199	-.474	-.001	.220	.050	.713	**.629**	**.836**	**.780**	.789	.770	7.50

Table 4: Comparison of various considered regression models and aggregation strategies for TAD (PRR↑, Gemma 7b model). Warmer colors indicate better results. The best method is in bold, the second best is underlined.

UQ Method	XSUM		PubMedQA		CoQA		Mean
	Rouge-L	AlignSc.	Rouge-L	AlignSc.	Acc.	AlignSc.	Rank
MSP	-.329	-.116	-.455	-.154	.699	**.626**	3.83
TAD Embeds. (LinReg, +AlignScore)	.191	.070	**.025**	.015	.606	.548	3.50
TAD Probs. (LinReg, +AlignScore)	.265	.234	-.360	-.142	**.712**	.613	2.83
TAD Attn. Only (LinReg+AlignScore)	.369	.252	-.345	-.112	.675	.608	2.67
TAD (LinReg+AlignScore)	**.541**	**.380**	.007	**.064**	.671	.600	**2.17**

Table 5: The comparison of various features for TAD (PRR↑, Gemma 7b model). The best method is in bold, the second best is underlined.

UQ Method	Runtime per batch	Overhead
MSP	10.26 ±2.78	—
Mean Token Entropy	10.29 ±2.79	0.26%
Focus	10.55 ±2.84	2.80%
EigValLaplacian	44.90 ±9.55	340%
MC SE	44.72 ±9.53	340%
Semantic Entropy	44.87 ±9.54	340%
SAR	57.63 ±12.57	460%
TAD (CatBoost)	10.34 ±2.80	0.80%
TAD (LinReg)	10.27 ±2.78	**0.10%**
TAD (MLP)	10.27 ±2.78	0.11%

Table 6: The evaluation of the runtime of UQ methods measured on 900 instances from all datasets with predictions from Llama 8b v3. The best results are in bold.

tention, but without probabilities. TAD trained on attention weights with probabilities substantially outperforms all other options. We also note that TAD trained only on embeddings performs much worse than other versions, which emphasizes the importance of both attention and probabilities.

Comparison to directly learning the unconditional probability. Table 13 compares TAD to directly learning the unconditional probability, where instead of using the target from Equation (2), we simply approximate $p(s_i = \mathrm{T})$. These results demonstrate that the attention weights contain a lot of information about the unconditional probability itself. Nevertheless, TAD's superior results show that taking into account the conditional dependency on previous generation steps and their uncertainty is also important.

4.4 Computational Efficiency

To demonstrate the computational efficiency of TAD, we compare its runtime to other UQ methods. We conducted experiments on 100 randomly sampled texts from each of our nine evaluation datasets using the LLaMA 8b v3 model on a single 80GB A100 GPU. The inference is implemented as a single-batch model call for all tokens in the output text. We use the LM-Polygraph (Fadeeva et al., 2023) implementation for other UQ methods.

Table 6 presents the average runtime per text sample for each UQ method, along with the percentage overhead over the standard LLM inference with MSP. As we can see, many state-of-the-art UQ methods such as (black-box, MC SE, Semantic Entropy, SAR) introduce huge computational overhead (340-460%) because they need to perform sampling from the LLM multiple times. On the contrary, TAD introduces minimal overhead (0.1-0.8%), which is much more practical.

5 Conclusion and Future Work

We have presented a new uncertainty quantification method based on learning conditional dependencies between the predictions made on multiple generation steps. The method relies on attention to construct features for learning this functional dependency and leverages this dependency to alter the uncertainty on subsequent generation steps. This yields improved results in selective generation tasks, especially when the LLM output is long. Our experimental study shows that our proposed technique usually outperforms other state-of-the-art UQ methods (such as SAR) resulting in the best overall performance across three LLMs and nine datasets. TAD does not introduce much computational overhead due to the simplicity of the regression model (linear regression), which makes it a potentially practical choice for LLM-based applications.

In future work, we aim to apply the suggested method to quantifying the uncertainty of retrieval-augmented LLMs. TAD potentially could be used to take into account the credibility of the retrieved evidence.

Limitations

In the motivation of our approach, we assume a strict Markov chain property between the generation steps. However, in reality, this property does not hold as the current generation step usually depends on multiple previous steps. This limitation of our method could be addressed by estimating the conditional dependency between multiple previous steps, e.g., by using a Transformer layer instead of the linear regressor. Nevertheless, our current implementation that makes the Markov assumption already yields strong results, and thus we leave investigation of more complex modifications for future work.

We also did not test our method on extra large LLMs such as LLaMA 3 70b. We only used 7-12b models due to limitations in our available computational resources.

Ethical Considerations

In our work, we considered open-source LLMs and datasets not aimed at harmful content. However, LLMs may generate potentially damaging texts for various groups of people. Uncertainty quantification techniques can help create more reliable use of neural networks. Moreover, they can be applied to detecting harmful generation, but this is not our intention.

Moreover, despite that our proposed method demonstrates significant performance improvements, it can still mistakenly highlight correct and not dangerous generated text with high uncertainty in some cases. Thus, as with other uncertainty quantification methods, it has limited application for various tasks.

References

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. *arXiv preprint arXiv:2307.15703*.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 245–255.

Julius Cheng and Andreas Vlachos. 2024. Measuring uncertainty in neural machine translation with similarity-sensitive entropy. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2115–2128, St. Julian's, Malta. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *Preprint*, arXiv:2307.01379.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*.

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. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

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

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A survey of language model confidence estimation and calibration. *arXiv preprint arXiv:2311.08298*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Jianfeng He, Linlin Yu, Shuo Lei, Chang-Tien Lu, and Feng Chen. 2024. Uncertainty estimation on sequential labeling via uncertainty transmission. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2823–2835, Mexico City, Mexico. Association for Computational Linguistics.

Jianfeng He, Xuchao Zhang, Shuo Lei, Zhiqian Chen, Fanglan Chen, Abdulaziz Alhamadani, Bei Xiao, and ChangTien Lu. 2020. Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8362–8372.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. 2022. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *CoRR*, abs/2305.19187.

Yu Lu, Jiali Zeng, Jiajun Zhang, Shuangzhi Wu, and Mu Li. 2022. Learning confidence for transformer-based neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2353–2364, Dublin, Ireland. Association for Computational Linguistics.

Andrey Malinin and Mark J. F. Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on Gemini research and technology. *CoRR*, abs/2403.08295.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*.

Yookoon Park and David Blei. 2024. Density uncertainty layers for reliable uncertainty estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 163–171. PMLR.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Junya Takayama and Yuki Arase. 2019. Relevant and informative response generation using pointwise mutual information. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 133–138. Association for Computational Linguistics.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 915–932, Singapore. Association for Computational Linguistics.

Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Mitigating uncertainty in document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3126–3136, Minneapolis, Minnesota. Association for Computational Linguistics.

A Additional Experimental Results

A.1 Comparison with other UQ Methods

Here, we present the main results with Llama and StableLM.

UQ Method	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
MSP	-.356	-.153	.358	.133	.002	.022	-.024	.033	.417	.493	**.324**	.174	.648	.557	.671	.590	.752	.706	7.17
Perplexity	-.388	-.124	-.088	.231	.130	.196	.239	-.023	.489	.513	.166	.129	.439	.413	-.456	-.457	.749	.696	8.44
Mean Token Entropy	-.385	-.124	-.114	.230	.132	.189	.233	-.035	.489	.509	.122	.119	.350	.353	-.498	-.481	**.756**	.708	9.11
Focus	-.356	-.110	-.024	.253	.112	**.201**	.045	-.063	.554	.540	.262	**.274**	.336	.261	-.469	-.377	.586	.587	9.50
NumSemSets	.011	.062	.154	.185	.070	.099	.005	.037	-.022	.098	.032	.168	.146	.288	.154	.232	.563	.657	10.78
DegMat	.048	.085	.191	.215	.076	.100	.013	.027	.069	.174	.112	.145	.306	.440	.317	.405	.633	.697	8.83
Eccentricity	-.009	.036	.034	.073	.042	.054	-.012	-.008	.048	.062	.086	.046	.484	.476	.386	.443	.643	.652	11.33
EigValLaplacian	.050	.086	.183	.217	.081	.100	.004	.029	.063	.172	.137	.166	.436	.478	.388	.450	.638	.687	7.94
Lexical Similarity	.011	.038	.302	.182	.105	.093	.099	.025	.272	.143	-.012	.012	.482	.473	.372	.414	.652	.647	9.22
MC NSE	-.058	.006	.216	.167	.117	.083	.070	-.006	.304	.217	.013	.012	.441	.407	.038	.071	.656	.637	10.72
MC SE	.029	.024	.253	.151	.071	.048	.029	.017	.101	.019	.134	.024	.511	.446	.425	.432	.633	.618	10.22
Semantic Entropy	.029	.026	.256	.157	.066	.050	.031	.015	.102	.022	.121	.023	.521	.483	.444	.459	.686	.675	9.06
SentenceSAR	-.095	-.005	.167	.125	.053	.033	-.028	.000	.033	.106	.203	.091	.584	.531	.547	.517	.729	.715	9.39
SAR	-.029	.038	.288	.208	.115	.112	.075	.012	.328	.237	.012	.085	.474	.489	.149	.181	.718	**.721**	7.78
TAD (LinReg)	.358	.223	.336	.219	**.210**	.111	**.429**	**.220**	.500	.501	.189	.130	.639	.561	.868	.758	.707	.671	3.72
TAD (LinReg, +AlignScore)	**.579**	**.345**	**.404**	**.369**	.207	.150	-.018	.083	**.613**	**.544**	.251	.235	**.657**	**.567**	**.914**	**.824**	.715	.691	**2.78**

Table 7: PRR↑ of UQ methods for the Llama 8b v3 model. Warmer colors indicate better results. The best method is in bold, the second best is underlined.

UQ Method	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
MSP	-.144	-.060	.498	.341	-.027	.062	-.429	-.168	.478	.596	.450	.224	.680	.597	.717	.685	.738	**.715**	6.83
Perplexity	-.257	-.034	.434	.351	.092	.044	.409	.099	.492	.592	.219	.179	.385	.373	-.340	-.385	.732	.700	8.33
Mean Token Entropy	-.250	-.028	.409	.340	.108	.034	.410	.085	.503	.593	.139	.181	.312	.329	-.403	-.423	**.747**	.713	9.00
Focus	-.173	.019	.300	.228	.040	.011	.214	.080	.559	.639	.217	.215	.147	.105	-.165	-.097	.643	.649	10.50
NumSemSets	.001	.054	.179	.187	.005	.074	.081	.051	-.007	.055	.060	.167	.221	.303	.110	.200	.576	.636	12.83
DegMat	-.000	.057	.309	.326	.017	.120	.052	.039	.136	.242	.214	.194	.342	.489	.452	.561	.653	.698	9.33
Eccentricity	-.034	.004	.235	.250	.023	.049	-.025	.007	.146	.179	.165	.047	.527	.557	.496	.568	.643	.660	11.61
EigValLaplacian	-.008	.063	.292	.311	.012	.115	.049	.038	.116	.226	.227	.215	.500	.557	.513	.581	.661	.697	8.78
Lexical Similarity	.111	**.079**	.381	.285	.119	.098	.094	.026	.296	.271	.141	.090	.508	.524	.489	.545	.656	.670	8.56
MC NSE	.068	.048	.371	.263	.073	.088	.161	.059	.370	.372	.123	.126	.437	.421	.273	.310	.623	.615	10.22
MC SE	.066	-.006	.393	.291	.059	.068	.034	.026	.209	.234	.164	.051	.565	.527	.515	.537	.623	.616	10.67
Semantic Entropy	.067	-.003	.412	.317	.066	.071	.033	.024	.215	.247	.152	.047	.578	.565	.545	.578	.674	.670	9.06
SentenceSAR	.005	.001	.392	.330	.010	.044	-.052	.001	.255	.307	.280	.157	.642	.603	.630	.644	.713	.713	8.22
SAR	.079	.079	.412	.341	.080	.119	.177	.059	.405	.401	.209	.196	.494	.531	.398	.460	.702	.714	6.33
TAD (LinReg)	.375	.024	.459	.282	**.163**	**.137**	**.493**	**.284**	.511	.610	.368	.222	**.707**	**.624**	.850	.786	.688	.671	3.50
TAD (LinReg+AlignScore)	**.459**	.068	**.519**	**.419**	.145	.127	.249	.219	**.696**	**.674**	**.462**	**.367**	.698	.614	**.863**	**.803**	.696	.691	**2.22**

Table 8: PRR↑ of UQ methods for the StableLM 12b v2 model. Warmer colors indicate better results. The best method is in bold, the second best is underlined.

A.2 Generalization Experiments

Tables 3, 9 and 10 present the comparison of the TAD trained on the in-domain training dataset with the TAD trained on all out-of-domain datasets for Gemma 7b, Llama 8b v3, and StableLM 12b v2 models respectively. In this experiment, we examine how our approach can be generalized on the unseen datasets. For each dataset, we create a general training dataset by using 300 samples from the training datasets from each of the eight other datasets used in the experiments. Thus, we evaluate TAD that is not trained on the target dataset. We conduct experiments on one dataset from each task: XSUM, PubMedQA, and CoQA. We compare the results with three strongest baseline methods: MSP, Focus, and SAR. Overall, we can see that the TAD method can be generalized on the unseen datasets and outperform all other baselines in most settings.

UQ Method	XSUM		PubMedQA		CoQA		Mean
	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	Acc.	AlignSc.	Rank
MSP	-.329	-.116	-.455	-.154	.699	**.626**	5.00
Focus	-.324	-.161	-.357	-.146	.322	.250	6.50
SAR	.042	-.006	.111	.014	.477	.503	4.50
TAD (LinReg)	.502	.257	**.576**	**.242**	.671	.608	**2.17**
TAD (LinReg+AlignSc.)	**.541**	**.380**	.007	.064	.671	.600	2.67
Gen. TAD (LinReg)	-.061	-.068	.288	.101	**.703**	.594	3.17
Gen. TAD (LinReg+AlignSc.)	.132	.096	-.124	-.074	.696	.589	4.00

Table 9: The comparison of TAD trained on in-domain data with TAD trained on all out-of-domain datasets (designated with "Gen.") (PRR↑, Gemma 7b model). Warmer colors indicate better results. The best method is in bold, the second best is underlined.

UQ Method	XSUM		PubMedQA		CoQA		Mean
	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Rank
MSP	-.144	-.060	-.429	-.168	.680	.597	5.83
Focus	-.173	.019	.214	.080	.147	.105	5.83
SAR	.079	**.079**	.177	.059	.494	.531	4.67
TAD (LinReg)	.375	.024	**.493**	**.284**	.707	**.624**	**1.67**
TAD (LinReg, +AlignScore)	**.459**	.068	.249	.219	.698	.614	2.50
Gen. TAD (LinReg)	-.032	-.015	.433	.217	.701	.584	4.00
Gen. TAD (LinReg, +AlignScore)	.023	-.008	.288	.143	**.709**	.592	3.50

Table 10: The comparison of TAD trained on in-domain data with TAD trained on all out-of-domain datasets (designated with "Gen.") (PRR↑, StableLM 12b v2 model). Warmer colors indicate better results. The best method is in bold, the second best is underlined.

A.3 Ablation Studies

Here, we present ablation studies for regression models and aggregation techniques with additional LLMs.

UQ Method	Aggregation	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
		ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.349	.183	-.064	.211	.180	.101	.366	.150	.448	.476	.208	.146	.605	.536	.741	.665	.743	.710	6.78
TAD (CatBoost, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.250	.097	-.013	.258	.137	**.192**	.255	.002	.448	.492	.234	.179	.576	.509	.605	.558	**.746**	**.714**	6.94
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	.357	.244	.279	.026	-.068	-.036	-.429	-.056	.293	.411	.323	.191	.647	.557	.813	.708	.715	.680	7.89
TAD (CatBoost, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.272	.227	.269	.015	-.115	-.050	-.461	-.070	.099	.239	.305	.189	**.672**	.566	.875	.795	.712	.683	8.61
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.358	.223	.336	.219	.210	.111	**.429**	.220	.500	.501	.189	.130	.535	.507	.742	.671	.739	.702	6.11
TAD (LinReg, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	**.579**	**.345**	**.404**	**.369**	.207	.150	-.018	.083	**.613**	**.544**	.251	.235	.509	.473	.637	.591	.738	.708	**4.89**
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	.438	.291	.307	.082	.005	-.021	-.402	-.049	.310	.421	**.396**	.261	.639	.561	.868	.758	.707	.671	6.61
TAD (LinReg, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.466	.334	.367	.098	-.040	-.041	-.447	-.065	.175	.256	.273	.195	.657	.567	**.914**	**.824**	.715	.691	6.11
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.496	.256	.317	.221	**.215**	.119	.408	.166	.509	.488	.189	.132	.587	.525	.751	.664	.738	.701	5.72
TAD (MLP, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.572	.326	.303	.346	.206	.145	.294	**.251**	.563	.487	.255	**.276**	.551	.494	.675	.635	.739	.712	**4.89**
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	.448	.303	.310	.069	.008	-.021	-.419	-.056	.301	.407	.355	.238	.646	.566	.879	.757	.718	.682	6.28
TAD (MLP, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.435	.326	.352	.088	-.052	-.046	-.453	-.063	.153	.220	.191	.146	.662	**.575**	.912	.822	.717	.693	7.17

Table 11: Comparison of various considered regression models and different aggregation strategies for TAD by PRR↑ for the Llama 8b v3 model for various tasks. Warmer colors indicate better results. The best method is in bold, the second best is underlined.

UQ Method	Aggregation	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Mean
		ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	ROUGE-L	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Acc.	AlignScore	Rank
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.374	.019	.409	.296	.117	.071	**.495**	.278	.500	.586	.394	.242	.637	.574	.710	.678	.725	.701	6.67
TAD (CatBoost, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.262	.036	.440	.311	.076	.060	.295	.107	.525	.589	.418	.297	.585	.529	.676	.650	**.726**	**.703**	6.94
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	.320	-.029	.442	.296	-.030	.151	-.565	-.188	.452	.586	**.539**	.236	.703	.619	.826	.763	.710	.675	7.39
TAD (CatBoost, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.248	-.021	.380	.255	-.105	.110	-.582	-.192	.316	.410	.451	.199	**.715**	.620	.862	.802	.702	.681	8.94
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.375	.024	.459	.282	.163	.137	.493	**.284**	.511	.610	.368	.222	.594	.555	.734	.710	.712	.686	5.83
TAD (LinReg, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.459	**.068**	**.519**	**.419**	.145	.127	.249	.219	.696	**.674**	.462	.367	.526	.488	.684	.661	.710	.693	5.06
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	.368	.011	.450	.279	.013	.154	-.556	-.185	.463	.599	.500	.228	.707	**.624**	.850	.786	.688	.653	6.50
TAD (LinReg, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.358	.036	.442	.324	-.023	.135	-.567	-.186	.429	.436	.453	.243	.698	.614	**.863**	**.803**	.696	.674	6.89
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	.401	.018	.473	.301	**.166**	.149	.488	.283	.516	.606	.397	.237	.605	.554	.728	.708	.711	.684	5.61
TAD (MLP, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	**.460**	.065	.500	.383	.135	.114	.249	.236	**.722**	.662	.525	**.414**	.528	.488	.733	.710	.705	.692	**4.67**
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	.363	.009	.458	.292	.024	**.169**	-.557	-.186	.475	.600	.507	.259	.701	.624	.843	.779	.695	.663	6.39
TAD (MLP, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	.343	.032	.441	.325	-.014	.136	-.571	-.186	.404	.396	.450	.227	.706	.621	.848	.790	.703	.684	7.11

Table 12: Comparison of various considered regression models and different aggregation strategies for TAD by PRR↑ for StableLM 12b v2 model for various tasks. Warmer colors indicate better results. The best method is in bold, the second best is underlined.

UQ Method	XSUM		SamSum		CNN		PubMedQA		MedQUAD		TruthfulQA		CoQA		SciQ		TriviaQA		Median
	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	ROUGE-L	AlignSc.	Acc.	AlignSc.	Acc.	AlignSc.	Acc.	AlignSc.	Acc.	AlignSc.	Rel. Impr.
Learning $p(s_i = T)$	.526	.345	.279	.314	**.182**	.079	-.014	.015	**.577**	.471	.460	**.389**	.657	.591	.809	.774	.743	.760	-
TAD (LinReg+AlignSc.)	**.541**	**.380**	**.353**	**.349**	.146	**.092**	**.007**	**.064**	.491	**.472**	**.505**	.368	**.671**	**.600**	**.834**	**.777**	**.784**	**.766**	**+3.1%**

Table 13: The comparison of TAD with directly learning the unconditional probability $p(s_i = \mathrm{T})$ (PRR↑, Gemma 7b model). The best method is in bold, the second best is underlined.

B Computational Resources

All experiments were conducted on a single NVIDIA A100 GPU. On average, training a single model across all datasets took over 750 GPU hours, while inference on the test set took 260 GPU hours.

C Hyperparameters

C.1 Optimal Hyperparameters for TAD

The optimal hyperparameters for TAD for various considered regression models and different aggregation strategies are presented in Tables 14 to 16 for Gemma 7b, Llama 8b v3, and StableLM 12b v2 models respectively. These hyperparameters are obtained using cross-validation with five folds using the training dataset. We train a regression model on $k-1$ folds of the training dataset and estimate uncertainty on the remaining fold. The optimal hyperparameters are selected according to the best average PRR for AlignScore. Finally, we use these hyperparameters to train the regression model on the entire training set.

The hyperparameter grid for the CatBoost is the following:
Num. of trees: [100, 200];
Learning rate: [1e-1, 1e-2];
Tree depth: [3, 5].

The hyperparameter grid for the linear regression is the following:
L2 regularization: [1e+1, 1, 1e-1, 1e-2, 1e-3, 1e-4].

The hyperparameter grid for the MLP is the following:
Num. of layers: [2, 4];
Num. of epochs: [10, 20, 30];
Learning rate: [1e-5, 3e-5, 5e-5];
Batch size: [64, 128].

UQ Method	Aggregation	XSUM	SamSum	CNN	PubMedQA	MedQUAD	TruthfulQA	CoQA	SciQ	TriviaQA
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 3	100, 0.01, 3	100, 0.01, 3	100, 0.01, 5	100, 0.1, 5	100, 0.01, 3	100, 0.01, 3	100, 0.01, 3	100, 0.01, 5
TAD (CatBoost, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 5	200, 0.01, 3	100, 0.01, 3	100, 0.1, 5	100, 0.01, 3	100, 0.01, 5	100, 0.01, 5	200, 0.1, 5	100, 0.01, 5
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 3	200, 0.1, 5	200, 0.1, 5	200, 0.01, 3	200, 0.1, 5	200, 0.1, 5	100, 0.1, 5	200, 0.1, 3	200, 0.1, 5
TAD (CatBoost, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 5	200, 0.1, 5	100, 0.01, 3	100, 0.1, 5	100, 0.01, 5	200, 0.1, 3	100, 0.01, 5	100, 0.01, 3	100, 0.01, 5
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	1	10.0	0.01	1	10.0	0.0001	10.0	10.0	10.0
TAD (LinReg, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	0.01	0.001	0.001	0.1	0.001	0.01	10.0	1	10.0
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	10.0	0.0001	0.0001	10.0	0.0001	0.01	10.0	1	1
TAD (LinReg, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	0.01	0.001	0.0001	0.0001	0.001	0.001	10.0	1	1
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	2, 30, 3e-05, 128	2, 10, 1e-05, 128	2, 30, 5e-05, 128	4, 10, 3e-05, 64	2, 10, 1e-05, 128	4, 30, 5e-05, 128	2, 10, 1e-05, 128	2, 10, 1e-05, 128	2, 10, 1e-05, 128
TAD (MLP, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	2, 30, 3e-05, 64	2, 30, 3e-05, 128	2, 30, 5e-05, 128	4, 10, 5e-05, 128	2, 10, 1e-05, 128	2, 10, 3e-05, 64	2, 10, 1e-05, 128	4, 10, 5e-05, 64	4, 10, 5e-05, 64
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	2, 20, 5e-05, 64	2, 10, 1e-05, 128	4, 30, 5e-05, 128	4, 10, 1e-05, 128	4, 30, 3e-05, 128	4, 20, 3e-05, 64	4, 10, 1e-05, 64	4, 30, 3e-05, 128	4, 30, 1e-05, 64
TAD (MLP, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	4, 20, 5e-05, 128	4, 30, 5e-05, 128	4, 20, 5e-05, 128	4, 30, 5e-05, 64	4, 30, 5e-05, 64	4, 30, 5e-05, 128	2, 20, 1e-05, 128	4, 20, 3e-05, 128	4, 30, 1e-05, 64

Table 14: Optimal hyperparameters for the TAD methods for the Gemma 7b model.

UQ Method	Aggregation	XSUM	SamSum	CNN	PubMedQA	MedQUAD	TruthfulQA	CoQA	SciQ	TriviaQA
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 5	100, 0.01, 3	200, 0.1, 5	100, 0.01, 3	100, 0.01, 3	200, 0.1, 5	100, 0.01, 5	100, 0.01, 3	100, 0.01, 3
TAD (CatBoost, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 5	200, 0.01, 3	100, 0.01, 3	100, 0.01, 5	200, 0.01, 5	200, 0.1, 5	100, 0.01, 5	200, 0.1, 5	100, 0.01, 5
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 5	100, 0.01, 3	200, 0.1, 5	200, 0.1, 5	200, 0.1, 5	200, 0.1, 5	200, 0.1, 3	200, 0.1, 3	200, 0.1, 3
TAD (CatBoost, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 5	100, 0.01, 3	100, 0.01, 5	100, 0.01, 5	100, 0.01, 5	200, 0.1, 5	100, 0.1, 3	200, 0.1, 5	100, 0.01, 5
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	0.0001	10.0	0.01	0.1	0.0001	10.0	10.0	10.0	10.0
TAD (LinReg, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	0.001	0.0001	0.01	0.01	0.0001	0.1	10.0	1	10.0
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	0.01	1	0.001	0.0001	0.0001	0.0001	10.0	1	10.0
TAD (LinReg, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	0.0001	0.0001	0.0001	0.1	0.0001	0.1	10.0	1	10.0
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	2, 10, 1e-05, 64	4, 30, 5e-05, 128	2, 30, 5e-05, 128	4, 10, 5e-05, 64	2, 20, 5e-05, 128	2, 30, 3e-05, 128	2, 10, 1e-05, 128	2, 30, 1e-05, 128	4, 30, 1e-05, 128
TAD (MLP, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	4, 10, 3e-05, 128	4, 20, 1e-05, 128	2, 20, 5e-05, 128	4, 10, 5e-05, 64	4, 30, 1e-05, 128	4, 30, 5e-05, 128	2, 10, 1e-05, 128	2, 20, 5e-05, 64	4, 10, 5e-05, 64
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	4, 10, 1e-05, 128	4, 10, 1e-05, 128	4, 30, 5e-05, 64	4, 20, 5e-05, 64	2, 30, 5e-05, 128	4, 30, 3e-05, 64	4, 10, 1e-05, 64	2, 10, 3e-05, 128	4, 20, 5e-05, 128
TAD (MLP, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	2, 30, 1e-05, 128	4, 30, 3e-05, 64	2, 30, 5e-05, 64	2, 20, 5e-05, 64	4, 30, 1e-05, 128	4, 30, 3e-05, 128	2, 10, 3e-05, 128	2, 30, 3e-05, 128	4, 10, 5e-05, 128

Table 15: Optimal hyperparameters for the TAD methods for the Llama 8b v3 model.

UQ Method	Aggregation	XSUM	SamSum	CNN	PubMedQA	MedQUAD	TruthfulQA	CoQA	SciQ	TriviaQA
TAD (CatBoost)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 5	100, 0.01, 3	100, 0.01, 3	200, 0.1, 5	200, 0.1, 5	200, 0.1, 3	100, 0.01, 3	100, 0.01, 3	200, 0.1, 3
TAD (CatBoost, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	200, 0.1, 3	100, 0.01, 3	100, 0.01, 5	200, 0.01, 3	100, 0.1, 3	200, 0.1, 5	100, 0.01, 5	200, 0.1, 5	100, 0.01, 5
TAD (CatBoost)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 5	100, 0.1, 3	200, 0.1, 5	200, 0.1, 3	200, 0.1, 5	200, 0.1, 5	100, 0.1, 5	100, 0.01, 5	200, 0.1, 3
TAD (CatBoost, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	200, 0.1, 3	100, 0.01, 3	100, 0.01, 5	200, 0.1, 5	100, 0.01, 5	200, 0.1, 5	100, 0.1, 3	100, 0.01, 5	200, 0.1, 3
TAD (LinReg)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	0.01	10.0	1	10.0	0.0001	0.0001	10.0	10.0	10.0
TAD (LinReg, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	0.01	0.1	0.1	0.0001	0.001	0.1	10.0	10.0	10.0
TAD (LinReg)	$\sum_{k=1}^{K} \log p_k$	1	10.0	1	1	0.01	0.001	10.0	10.0	10.0
TAD (LinReg, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	0.1	0.001	0.01	1	0.1	0.001	10.0	10.0	10.0
TAD (MLP)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	4, 10, 1e-05, 128	4, 10, 5e-05, 64	2, 30, 1e-05, 128	2, 30, 1e-05, 128	4, 30, 3e-05, 64	4, 30, 1e-05, 128	2, 10, 1e-05, 128	4, 10, 3e-05, 64	4, 30, 1e-05, 128
TAD (MLP, +AlignScore)	$\frac{1}{K}\sum_{k=1}^{K} p_k$	2, 30, 3e-05, 128	4, 10, 1e-05, 128	4, 30, 5e-05, 64	4, 10, 1e-05, 128	4, 10, 5e-05, 128	4, 30, 1e-05, 64	2, 10, 3e-05, 128	4, 10, 3e-05, 64	4, 10, 3e-05, 64
TAD (MLP)	$\sum_{k=1}^{K} \log p_k$	2, 20, 1e-05, 128	4, 10, 5e-05, 64	4, 30, 5e-05, 64	4, 20, 5e-05, 64	4, 20, 5e-05, 64	4, 20, 5e-05, 64	4, 10, 3e-05, 64	2, 10, 1e-05, 64	4, 10, 3e-05, 64
TAD (MLP, +AlignScore)	$\sum_{k=1}^{K} \log p_k$	2, 30, 1e-05, 64	2, 30, 3e-05, 64	4, 30, 5e-05, 64	4, 20, 1e-05, 64	4, 30, 3e-05, 128	4, 30, 5e-05, 128	2, 10, 1e-05, 64	4, 10, 1e-05, 128	4, 10, 3e-05, 64

Table 16: Optimal hyperparameters for the TAD methods for the StableLM 12b v2 model.

C.2 LLM Generation Hyperparameters

Dataset	Task	Max Input Length	Generation Length	Temperature	Top-p	Do Sample	Beams	Repetition Penalty
XSum			56					
SamSum	TS		128					
CNN			128					
PubMedQA			128					
MedQUAD	QA Long answer	-	128	1.0	1.0	False	1	1
TruthfulQA			128					
CoQA			20					
SciQ	QA Short answer		20					
TriviQA			20					

Table 17: Text generation hyperparameters for all LLMs used in the experiments.

D Dataset Statistics

Task	Dataset	N-shot	Train texts for TAD	Evaluation texts
Text Summarization	CNN/DailyMail	0	2,000	2,000
	XSum	0	2,000	2,000
	SamSum	0	2,000	819
QA Long answer	PubMedQA	0	2,000	2,000
	MedQUAD	5	1,000	2,000
	TruthfulQA	5	408	409
QA Short answer	SciQ	0	2,000	1,000
	CoQA	all preceding questions	2,000	2,000
	TriviaQA	5	2,000	2,000

Table 18: The statistics of the datasets used for evaluation.