A Head to Predict and a Head to Question: Pre-trained Uncertainty Quantification Heads for Hallucination Detection in LLM Outputs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have the tendency to hallucinate, i.e., to sporadically generate false or fabricated information. This presents a major challenge, as hallucinations often appear highly convincing and users generally lack the tools to detect them. Uncertainty quantification (UQ) provides a framework for assessing the reliability of model outputs, aiding in the identification of potential hallucinations. In this work, we introduce pre-trained UQ heads: supervised auxiliary modules for LLMs that substantially enhance their ability to capture uncertainty compared to unsupervised UQ methods. Their strong performance stems from the powerful Transformer architecture in their design and informative features derived from LLM attention maps. Experimental evaluation shows that these heads are highly robust and achieve state-of-the-art performance in claim-level hallucination detection across both in-domain and out-of-domain prompts. Moreover, these modules demonstrate strong generalization to languages they were not explicitly trained on. We pre-train a collection of UQ heads for popular LLM series, including Mistral, Llama, and Gemma 2. We publicly release both the code and the pre-trained heads.[1]

1 Introduction

Uncertainty quantification (UQ) (Gal and Ghahramani, 2016; Baan et al., 2023; Geng et al., 2024; Zhang et al., 2024a) has become an increasingly important topic in natural language processing (NLP), particularly for addressing challenges with hallucinations (Huang et al., 2025) and low-quality outputs of large language models (LLMs) (Malinin and Gales, 2021; Kuhn et al., 2023; Fadeeva et al., 2024). UQ offers the potential to improve the safety and reliability of LLM-based applications by flagging highly uncertain generations. Such generations could be discarded or marked as untrustworthy, thus reducing the risk of misleading information reaching users (Zhang et al., 2024a,b; Huang et al., 2024). Contrary to other methods for detecting hallucinations that rely on external knowledge bases or additional LLMs (Manakul et al., 2023; Min et al., 2023; Chen et al., 2023), UQ methods assume that LLMs naturally encode information about their own limitations, and this self-knowledge can be efficiently accessed.

There are many existing UQ techniques for well-defined tasks such as classification and regression (Zhang et al., 2019; He et al., 2020; Xin et al., 2021; Wang et al., 2022; Vazhentsev et al., 2023; He et al., 2024a). However, applying UQ to text generation has unique challenges including (*i*) an infinite number of possible generations, which complicates the normalization of the uncertainty scores, (*ii*) potentially multiple correct answers with different surface forms (Kuhn et al., 2023), (*iii*) need to aggregate uncertainties across multiple conditionally dependent predictions (Zhang et al., 2023), (*iv*) generated tokens not contributing to uncertainty equally (Duan et al., 2024), and (*v*) some sources of uncertainty being irrelevant for hallucination detection (Fadeeva et al., 2024). These challenges hinder the performance of classical unsupervised UQ techniques, as they are difficult to address explicitly within a single method. Recently, researchers have proposed learning the aforementioned intricacies from the annotated data and developed supervised methods for UQ and hallucination detection (Azaria and Mitchell, 2023; Li et al., 2024; He et al., 2024b; Chuang et al., 2024).

We continue this line of work by introducing pre-trained UQ heads: supervised auxiliary modules for LLMs that substantially enhance their ability to capture uncertainty compared to unsupervised UQ methods. Their strong performance stems from the powerful Transformer architecture in their design

---

[1] `https://anonymous.4open.science/r/llm-uncertainty-head-24DD`

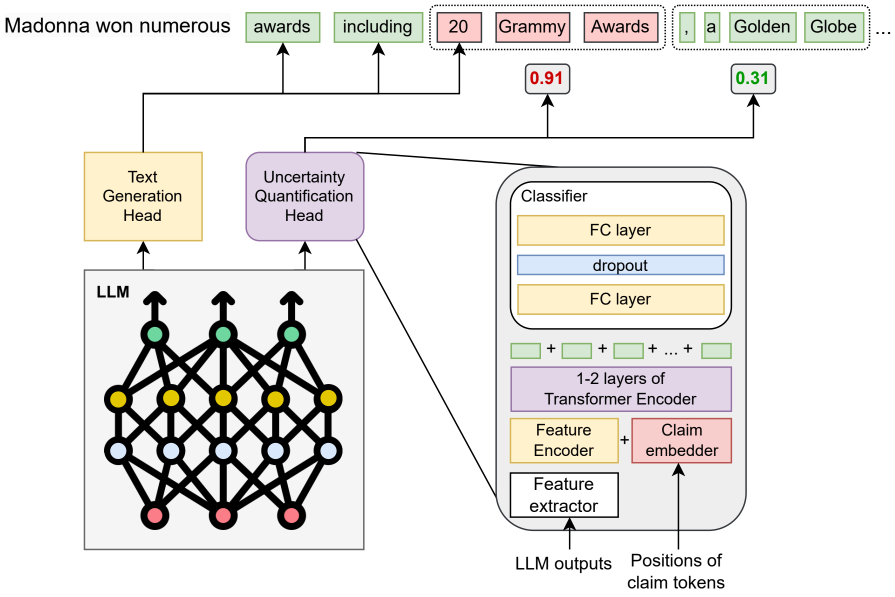


Figure 1: The architecture of uncertainty quantification heads. The example represents a text generated using an LLM, containing the hallucination *20 Grammy Awards* highlighted in red.

and informative features derived from LLM attention maps. These heads do not require re-training of the entire LLM and do not alter its outputs. In addition to their high performance, these methods maintain a relatively small memory and computational footprint, ensuring practical usability.

Experimental evaluation shows that our uncertainty heads are highly robust and achieve state-of-the-art performance in claim-level hallucination detection across both in-domain and out-of-domain prompts, outperforming other supervised and unsupervised techniques. Moreover, these modules demonstrate strong generalization to languages they were not explicitly trained on.

Training UQ heads requires annotated hallucinations in LLM outputs. For constructing training data, we created an automatic annotation pipeline, which allowed us to scale experiments and to pre-train UQ heads for various LLMs. We release a collection of pre-trained UQ heads for popular open-source instruction-following LLMs, including Llama series (Dubey et al., 2024), Gemma 2 (Team et al., 2023), and Mistral (Jiang et al., 2023a). The **contributions** of this work are as follows:

- We design a pre-trained uncertainty quantification head: a supplementary module for an LLM that yields substantially better performance for claim-level hallucination detection than classical unsupervised UQ methods and state-of-the-art supervised techniques.
- We conduct a vast empirical investigation and find that uncertainty heads show good generalization across various domains and languages. We perform a vast ablation study that compares various feature sets architectures, and approaches to training data generation.
- We build and release a collection of pre-trained UQ heads for popular series of open-source instruction-tuned LLMs. These modules could be seamlessly integrated into text generation code and be used as off-the-shelf hallucination detection tools.

2 Related Work

Unsupervised UQ methods for LLMs can be broadly categorized into five groups: information-based approaches (Kuhn et al., 2023; Farquhar et al., 2024), density-based scores (Vazhentsev et al., 2022; Ren et al., 2023), self-consistency methods (Lin et al., 2023; Zhang et al., 2024a; Qiu and Miikkulainen, 2024), methods grounded in mechanistic analysis of LLMs (Yuksekgonul et al., 2023; Qiu and Miikkulainen, 2024), and verbalized (reflexive) strategies (Kadavath et al., 2022; Tian et al., 2023). Although they have demonstrated potential, their effectiveness in hallucination detection remains limited (Vashurin et al., 2024).

Supervised UQ methods leverage the internal states of LLMs as features for predicting hallucinations (Azaria and Mitchell, 2023; Slobodkin et al., 2023; Su et al., 2024; CH-Wang et al., 2024; He et al., 2024b; Chuang et al., 2024). These recently emerged methods achieve substantial performance gains over unsupervised approaches, especially for in-domain data.

Azaria and Mitchell (2023) proposed one of the first methods of this kind called SAPLMA, where they trained a perceptron using layer activations to detect when a LLM "agrees" with false statements. Slobodkin et al. (2023) trained a linear model on hidden states to detect question "answerability", effectively identifying unanswerable questions that typically lead to hallucinations. CH-Wang et al. (2024) extend this approach to span-level hallucination detection, using manually annotated hallucinations. He et al. (2024b) experiment with activation maps, token ranks, and probabilities from unembedding matrices across layers. Chuang et al. (2024) introduces a feature set derived from LLM attention maps.

Limitations of previous methods. Azaria and Mitchell (2023); Slobodkin et al. (2023); Su et al. (2024) focused on sequence-level methods and are not able to detect sub-sentence hallucinations. Many models, including Slobodkin et al. (2023); Azaria and Mitchell (2023); Chuang et al. (2024); Su et al. (2024) used non-contextualized architectures such as simple linear probes or multi-layer perceptron. Although He et al. (2024b) integrated a linear model with an attention mechanism and CH-Wang et al. (2024) used a contextualized model combining convolutions, ResNet, and GRU, these architectures are considered outdated and exhibit limitations in quality or computational efficiency. The features of the majority of models included only hidden states (Azaria and Mitchell, 2023; Slobodkin et al., 2023; CH-Wang et al., 2024; Su et al., 2024), which limits their generalization. Only He et al. (2024b) and Chuang et al. (2024) performed more elaborate feature engineering. Finally, synthetic data that is leveraged through enforced decoding is used in some work (Azaria and Mitchell, 2023; Slobodkin et al., 2023). Compared to the native outputs generated by LLMs, such data may introduce additional biases and adversely affect the performance of hallucination detectors.

In contrast to most prior work, we focus on building UQ heads specifically for detecting hallucinations at the subsentence level, i.e., individual atomic claims. Our approach leverages the strengths of previous methods while addressing their key limitations: (*i*) instead of outdated architectures, we build our solution on the powerful Transformer architecture, (*ii*) we investigate the importance of various features for hallucination detection, finding that the most informative features are derived from attention maps of LLMs, and (*iii*) we build an automatic pipeline for generating training data using the *native* LLM responses. This pipeline allows us to build training data at a larger scale and pre-train UQ heads for a range of popular LLMs.

3 Uncertainty Quantification Head

Consider the LLM $P(t_i \mid \mathbf{x}, \mathbf{t}_{<i})$ with L layers receiving a prompt $\mathbf{x}$ of length n and generating tokens $\mathbf{t} = \{t_1, t_2, ..., t_T\}$. We also have a set of atomic claims $C = \{c_1, c_2, ..., c_K\}$, each representing a mapping to a subset of tokens in the output. Atomic claims, for example, can be extracted by another lightweight model. In this work, we formalize the claim-level uncertainty quantification task as building a function $U(c_i|\mathbf{x}, \mathbf{t}) \in [0, 1]$ that determines whether the claim $c_i \in C$ is a hallucination. A large value of $U(c_i|\mathbf{x}, \mathbf{t})$ indicates a higher likelihood that the claim c_i is a hallucination.

3.1 Background on Features for UQ and Hallucination Detection

Hidden states $h(t)$ extracted from LLM layers during the generation of a token t have been shown to serve as strong indicators of hallucinations in several studies (Azaria and Mitchell, 2023; CH-Wang et al., 2024).

$$F_{\text{hs}}(t) = h(t) \tag{1}$$

Lookback Lens (Chuang et al., 2024) derives features from the LLM's attention maps. The key idea is that when the model attends to the prompt, it attempts to solve the task, whereas attending to generated tokens causes it to disregard the prompt, increasing the likelihood of hallucination. They suggest using the so-called lookback ratio – the ratio of aggregated attention to tokens of the prompt and the generated tokens. Consider each layer of the LLM contains Q attention heads, q is an index of a head, and $\alpha^{h,l}_{t_i,t_j}$ represents the softmax-weighted attention score from token t_i to token t_j. $A^{q,l}_{\text{context}}(t_i)$ and $A^{q,l}_{\text{gen}}(t_i)$ are the average attention weights to the input $\mathbf{x}$ and to the previously generated output $\mathbf{t}_{<i}$, respectively:

$$A^{q,l}_{\text{context}}(t_i) = \frac{1}{n}\sum_{j=1}^{n}\alpha^{q,l}_{t_i,x_j},$$

$$A^{q,l}_{\text{gen}}(t_i) = \frac{1}{i-1}\sum_{j=n+1}^{i-1}\alpha^{q,l}_{t_i,t_j}.$$

Then the lookback ratio of the model head q and the layer l for the token t_i is defined as follows:

$$LR^{q,l}(t_i) = \frac{A^{q,l}_{\text{context}}(t_i)}{A^{q,l}_{\text{context}}(t_i) + A^{q,l}_{\text{gen}}(t_i)},$$

$$F_{\text{LBLens}}(t_i) = \{LR^{q,l}(t_i)\}_{q,l}^{Q,L}. \tag{2}$$

Factoscope (Min et al., 2023) in addition to model activations, uses a set of features that leverage token probabilities, the similarity of token embeddings across layers, and the evolution of token ranks across layers. Commonly, given a token t_i at the position i, the LLM outputs hidden states $\{h_l(t_i)\}_{l=1}^{L}$, where the final hidden state $h_L(t_i)$ is passed through the unembedding matrix E to predict token logits. Factoscope applies E to each LLM layer, obtaining a set of token logits on a specific layer l: $z_i^l = E\left(h_l(t_i)\right)$. Then, it extracts the logits of the top-m tokens from each layer l:

$$F_{\text{top-tokens}}(t_i) = \left\{z_i^l(t) \mid t \in \text{top}_m(z_i^l)\right\}_{l=1}^{L}. \tag{3}$$

To analyze token evolution across layers, Factoscope computes the cosine similarities between embeddings of top tokens from adjacent layers obtained by applying the unembedding matrix:

$$\begin{aligned} S^l(t_i) = \{\, &\cos(E_{w_1}, E_{w_2}) \mid \\ &w_1 \in \text{top}_m(z_i^l), w_2 \in \text{top}_m(z_i^{l+1})\} \end{aligned}$$

$$F_{\text{tokens-sim}}(t_i) = \{S^l(t_i)\}_{l=1}^{L-1}. \tag{4}$$

Finally, Factoscope tracks token rank evolution across layers: $R^l(t_i) = \text{rank}[t_i, z_i^l]$, where rank indicates the position of t_i in the descending order of z_i^l values (top-ranked token receives 1). The ranks are further normalized to the range $[0, 1]$:

$$F_{\text{rank}}(t_i) = \{R^l(t_i)^{-1}\}_{l=1}^{L}. \tag{5}$$

3.2 Features for Pre-trained UQ Heads

We experimented with all the aforementioned types of features and their combinations. However, we found that all of them exhibited various limitations. Hidden states encode a lot of domain-specific information, increasing the risk of overfitting. Factoscope features incur substantial computational overhead while offering limited additional information beyond what is captured by hidden states. Attention features are quite powerful, but the aggregation suggested in Lookback Lens results in the loss of valuable information. Moreover, they underperform without the addition of logits or probabilities. Therefore, for our pre-trained UQ heads, we use two groups of features.

Attention maps of the LLM. Mechanistic analysis of attention weights reveals that attention patterns often reflect the model's behavior under uncertainty (Yuksekgonul et al., 2023). Moreover, attention encodes the conditional dependency between the generation steps (Zhang et al., 2023). For each token, we obtain the attention maps to k previous tokens from each attention head and layer and flatten them into a single feature vector without aggregation:

$$F_{\text{att}}(t_i) = \{\alpha^{q,l}_{t_i,t_{i-j}}\}_{j,q,l}^{k,Q,L}. \tag{6}$$

When $(i-j)$ is negative, we pad the feature vector with zero placeholders. While considering many previous tokens might explode the feature space size, we empirically found that the optimal value of k is typically very small: $1 \leq k \leq 5$ (see Figure 5). As a contextualized architecture, the Transformer can automatically extract meaningful patterns across the entire generated sequence without requiring explicit features from previous tokens.

Probability distribution of the LLM might be misleading, but it still conveys useful information about the model's conditional confidence at the current generation step. This group of features consists of logarithms of the top-m token probabilities:

$$\begin{aligned} F_{\text{prob}}(t_i) = \{\, &\log P(t \mid \mathbf{x}, \mathbf{t}_{<i}) \mid \\ &t \in \text{top}_m(P(\cdot \mid \mathbf{x}, \mathbf{t}_{<i}))\}. \end{aligned} \tag{7}$$

Features from both groups are concatenated into a token-level vector: $F(t) = F_{\text{att}}(t) \circ F_{\text{prob}}(t)$.

3.3 Architecture of UQ Heads

The architecture of the UQ head is depicted in Figure 1. To ensure flexibility and expressive capacity, we build it on top of a Transformer backbone. It consists of a feature size reduction network, a multi-layer transformer encoder, and a two-layer classification neural network. For each component, we use GELU activation functions and dropout regularization. To mark tokens as belonging to the classified claim, we introduce an embedding matrix. Each token, depending on whether it belongs to the classified claim, receives a corresponding embedding that is summed up with the representation from the feature size reduction network. The resulting representations are fed into the transformer encoder. The outputs of the encoder are averaged across all tokens of the claim and fed into the classifier. The UQ head is trained using a binary cross-entropy loss function. When we train heads, we freeze the "body" of the LLM, so that the LLM generations stay exactly the same.

4 Pipeline for Training Data Generation

The training data generation pipeline is presented in Figure 4 in the appendix. It starts with prompting the LLM to produce responses for a list of questions such as *Write a biography of person X* or *Write the history of the city Y*. We select relatively famous named entities so the task is not very hard for the model based on its parametric knowledge, while at the same time, it is not trivial, so outputs contain some hallucinated claims. We also do not use synthetically-generated hallucinations, as they introduce a bias between what the model actually generates vs. the synthetic data. The prompts for other domains can be found in Table 5.

We split the obtained responses into atomic claims using GPT-4o using the prompts from (Fadeeva et al., 2024; Vashurin et al., 2024). Each claim is then automatically classified by GPT-4o as *supported*, *unsupported*, or *unknown*. The last category is intended for general claims, for which estimating the veracity is meaningless. To ensure high annotation quality, the claim labeling process is two-staged: first, we ask the model to provide an elaborated answer via chain-of-thought; then, we ask it to summarize its answer into one word. As shown in prior studies (Vashurin et al., 2024) the performance of such annotation using GPT-4o is high (accuracy over 90%). However, it could potentially be further improved by leveraging more powerful LLMs or employing model ensembles.

The pipeline enables the cost-effective construction of large datasets annotated with claim-level hallucinations across various LLMs. The cost of annotating responses from a single LLM on the training biographies dataset, consisting of 3,300 prompts, was approximately $100. Statistics about the training dataset, as well as the accuracy of LLM responses, are presented in Table 4.

5 Experiments

5.1 Experimental Setup

Evaluation datasets. We constructed eight test sets of English questions designed to prompt LLMs to generate text across various domains: *person biographies*, *cities*, *movies*, *inventions*, *books*, *artworks*, *landmarks*, and *events*. Each test set contains 100 questions, generated by prompting GPT-4o and Claude-3-Opus to output 100 famous domain items, e.g., 100 famous landmarks. Examples of the prompts are presented in Appendix B.1.[2] The labels for the test sets are obtained using the same annotation pipeline as for the training data.

To assess the cross-lingual generalizability of pre-trained UQ modules, we conducted evaluations on Russian and Chinese prompts from (Vashurin et al., 2024), and additionally created a similar test set with German prompts. Test sets for each language consist of 100 biography-related questions. The data statistics are presented in Table 5.

Metrics. In the main experiments, we measured the claim-level performance of detecting invalid claims. For this purpose, we used PR-AUC, where "unsupported" claims represent the positive class.

Models. We conducted our primary experiments with Mistral 7b Instruct v0.2 (Jiang et al., 2023b) and Gemma 2 9b Instruct (Team et al., 2023).

Training procedure and hyper-parameter optimization. We trained the uncertainty heads using Adam with a linear learning rate decay and warmup. We selected the values of the hyper-parameters on the validation set of the *biographies* dataset using the claim-level PR-AUC metric and the Bayesian optimization algorithm available in the W&B framework. We observed that among important hyper-parameters are the weight of instances with positive labels, the number of epochs, and the learning rate. The best values of hyper-parameters for each of the tested models are presented in Table 11 in Appendix F.

Baselines. We compare our method to several unsupervised baselines: Maximum Claim Probability (an adaptation of Maximum Sequence Probability for claims), Mean Token Entropy, Perplexity, Claim Conditioned Probability (CCP) (Fadeeva et al., 2024), and Attention Score (Qiu and Miikkulainen, 2024). Furthermore, we evaluated

---

[2] All data used for training and testing is available at <anonymized>

Test Sets **Method**	**Biographies (in domain)**	**Cities**	**Movies**	**Inventions**	**Books**	**Artworks**	**Landmarks**	**Events**
Random	.291	.205	.099	.163	.110	.264	.117	.113
MCP	.412	.310	.205	.319	.145	.317	.135	.141
Perplexity	.361	.231	.170	.232	.138	.335	.128	.123
Max Token Entropy	.416	.289	.241	.381	.171	.321	.141	.161
Attention Score	.333	.279	.114	.211	.114	.202	.125	.132
CCP	.496	.368	.267	.380	.167	.382	.196	.171
SAPLMA	.536	.435	.269	.350	.292	.534	**.350**	.235
Factoscope	.611	.468	.344	.424	.315	.485	.279	.265
Lookback Lens	.557	.449	.254	.391	.259	.464	.257	.295
UHead (Ours)	**.660**	**.487**	**.466**	**.485**	**.395**	**.561**	.340	**.369**

Table 1: PR-AUC for various UQ methods for hallucination detection of the Mistral 7B Instruct v0.2 model on English datasets. Biographies represent the in-domain dataset for supervised UQ methods.

Language **Method**	**English (in domain)**	**Russian**	**Chinese**	**German**
Random	.133	.337	.226	.152
MCP	.180	.433	.307	.203
Perplexity	.136	.395	.287	.149
Max Token Entropy	.202	.437	.444	.217
Attention Score	.146	.446	.230	.229
CCP	.307	.493	.439	.306
SAPLMA	.342	.514	.331	.391
Factoscope	.354	.532	.350	.380
Lookback Lens	.359	.576	.479	.390
UHead (Ours)	**.457**	**.581**	**.556**	**.455**

Table 2: PR-AUC of UQ methods on various languages using the Gemma 2 9b Instruct model. Supervised detectors were trained on English-only *biographies* data.

Test Set **Method**	**Biographies (dev)**
UHead (only hidden states)	.582
UHead (att. + probs. + hs.)	.589
UHead (Factoscope)	.588
UHead (LookBack Lens)	.609
UHead (att.)	.617
UHead (att. + probs.) (ours)	**.642**

Table 3: PR-AUC scores for UQ heads trained with various feature sets on the Mistral 7B Instruct v0.2 model. Performance was evaluated using the validation set of the *biographies* domain after hyperparameter tuning.

our UQ heads against supervised methods, including SAPLMA, Factoscope, and Lookback Lens. SAPLMA predicts token-level uncertainties using a 3-layer perceptron, and the mean uncertainty is calculated over claim-related tokens during inference. We adapt Lookback Lens and Factoscope to the claim level. Lookback Lens uses a Logistic Regression model trained on lookback ratios. Our implementation of Factoscope uses our transformer-based architecture and the feature set that includes hidden states, top token embeddings with similarities, and token ranks. The values of the hyperparameters for the baselines selected after tuning are given in Appendix F.

5.2 Results

Main results. Table 1 shows the performance of the unsupervised UQ techniques and the supervised detectors trained on persons' *biographies* for claim-level hallucination detection with Mistral 7B Instruct v0.2. For evaluating supervised methods, the domain *biographies* represents the in-domain test set and all other domains represent out-of-domain (OOD) test sets. Note that in this evaluation, both the questions and the LLM's responses across all domains are in English.

Among the unsupervised techniques, uncertainty scores based on CCP yield the best performance, confidently outperforming other methods on *biographies*, *cities*, *artworks*, and *landmarks*.

Supervised UQ methods greatly outperform unsupervised techniques on the in-domain test set. Moreover, remarkably, all considered supervised methods demonstrate substantial generalization and the ability to perform well beyond the training domain of people's *biographies*.

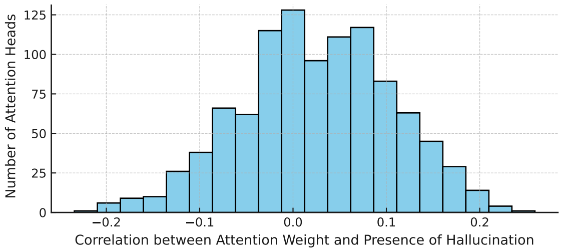
Our UQ head (UHead) demonstrates the best results in both in-domain and out-of-domain evaluations. For in-domain evaluation, UHead outperforms the best unsupervised method CCP by 16 percentage points (pp) in terms of PR-AUC. The gap is also large for out-of-domain evaluation, e.g., for *books*, UHead outperforms CCP by 23 pp, for *movies* and *events* by 20 pp, for *artworks* by 18 pp. Table 7 in Appendix E also shows that UHead pre-trained on *biographies* generalizes to question answering on the TruthfulQA dataset (Lin et al., 2022), outperforming unsupervised baselines.

When evaluated alongside supervised methods, UHead surpasses the closest competitor, Factoscope, by five pp for the in-domain evaluation. In OOD evaluation, it confidently outperforms other supervised methods across all domains, except for *landmarks*, where it is slightly below the closest competitor by 1 pp.

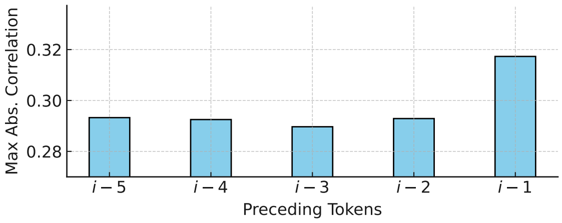
Analyzing other supervised methods, the second-best scores are usually demonstrated by Factoscope. We assume that the underperformance of the baseline based on the Factoscope features compared to UHead lies in the use of layer activations, which limit its generalization. Another module that relies on hidden states is SAPLMA. In addition to the feature limitations, it also has architectural limitations, which further hurt its performance. For *landmarks*, SAPLMA shows good results, but for other test sets, it stays behind Factoscope and UHead. Compared to UHead, it lags by 12 pp on in-domain evaluation and up to 20 pp on OOD evaluation. Lookback Lens also usually falls behind UHead and Factoscope; we believe that its main problem is its weak linear architecture.

Cross-lingual generalization. Table 2 presents the cross-lingual results for Gemma 2 9b Instruct. In this experiment, we train UQ modules on the English person's *biographies* as in the previous experiment, but we evaluate the performance on other languages. Surprisingly, UHead achieves strong cross-lingual generalization. For all OOD languages, UHead achieves substantial improvements over the best unsupervised methods. For Chinese, UHead is better than MTE by 10 pp; for Russian, it is better than CCP by 9 pp; and for German by 13 pp. Notably, other supervised methods also demonstrate some level of generalization, but in most cases, they have substantially worse performance. Overall, these results show that UQ heads, even if they are pre-trained on English data, can serve as effective off-the-shelf hallucination detectors for LLM outputs in other languages.

Analysis of feature sets. Table 3 presents the comparison of various feature sets in combination with the UHead architecture on the in-domain validation set. For each feature set, we perform an extensive hyper-parameter value search in the same way as for the main results. We can see that all feature sets that leverage hidden states fall substantially behind attention-based features. The analysis of the validation loss dynamics shows that this is probably due to quick overfitting. Models that leverage hidden states start overfitting after 1–3 epochs, while models that leverage attention might not overfit even after 10 epochs. We also note that Lookback Lens features combined with the UHead architecture provide strong performance. However, simple attention maps without feature engineering used in UHead yield even better results. Finally, without probability-based features, UHead loses around 2.5 pp PR-AUC, which marks their importance.



(a) Correlation across various attention heads.



(b) Max absolute attention–hallucination correlation.

Figure 2: (a) The distribution of correlations between attention on the $i-1$-th token and presence of the i-th token in hallucinated claim. (b) The maximum absolute correlation across heads and layers for the same phenomenon. All scores were computed using the Mistral model and the *biographies* dataset.

Analysis of attention-based features. We examined attention patterns that may serve as indicators of hallucinations in generated tokens. Figure 2a shows the correlation between the presence of hallucination and attention weights from the generated token to the immediately preceding token across various attention heads. While most heads show negligible correlation, a subset of heads exhibits moderate positive or negative associations. Figure 2b further highlights that this correlation is strongest for the token immediately preceding the generated one. Thus, a small subset of attention heads encodes informative signals related to hallucinations and reflects distinct model behavior under uncertainty during generation.

These findings are also confirmed by Figures 5 and 6. Figure 6 illustrates that attention weights from individual middle layers could serve as relatively strong hallucination detectors. Figure 5 shows that optimal performance is obtained by UHead when using attention weights from only 1–5 preceding tokens.

Analysis of detector architectures. Table 8 in

Appendix E reports the performance of detectors with various architectures trained on our best feature set, consisting of attention maps and top token probabilities. We compare the transformer-based architecture used in UHead against simpler alternatives: MLP and a linear model. Although both simpler models yield notable improvements over the best unsupervised baselines, UHead based on transformer achieves the highest performance.

Introducing more diverse training data for UHead. Table 10 presents the results when we train uncertainty heads on *biographies* plus the data from all domains except one, which is used for OOD evaluation. In this scenario, uncertainty heads get access to bigger and more diverse training data. As we can see, expanding the dataset provides slight improvements for certain domains. These results indicate that expanding the training data and enhancing its diversity could further increase the UQ performance, particularly in the OOD setting.

Using "non-native" training data. We also analyzed the possibility of using the training data generated for one LLM for training a detector for another LLM. We take the annotated dataset generated by Mistral and performed inference of Gemma 2 via forced decoding to generate features. Table 9 in Appendix E compares the results of hallucination detectors for Gemma 2 trained using "native" and "non-native" data. We can see that "non-native" data drastically decreases the performance. Therefore, for each new LLM, we recommend generating a new training dataset.

Computational efficiency. We evaluated the computational overhead of various UQ methods. To ensure a fair comparison, we focused only on the time required to generate texts and to compute uncertainty scores, excluding the time spent on claim extraction. Claim extraction could be performed by a small model specifically fine-tuned for this task, and its overhead is negligible compared to LLM inference. Table 6 summarizes the results and provides the memory footprint of methods. MCP and Perplexity incur no additional overhead, serving as baselines for comparison. Our UHead introduces only 5% overhead, which is even better than the best unsupervised method CCP (8.6%). UHead also has a minimal GPU memory footprint (40 MB). Thus, UHead is a very lightweight addition to multi-billion-parameter LLMs and is practical for real-world deployment.

```
from transformers import AutoModelForCausalLM,
     AutoTokenizer
from luh import AutoUncertaintyHead,
     CausalLMWithUncertainty

llm = AutoModelForCausalLM.from_pretrained(
    model_name)
tokenizer = AutoTokenizer.from_pretrained(
    model_name)
uhead = AutoUncertaintyHead.from_pretrained(
    uhead_name, base_model=llm)
llm_adapter = CausalLMWithUncertainty(llm, uhead,
     tokenizer=tokenizer)

# tokenize text and prepare inputs ...
output = llm_adapter.generate(inputs)
```

Figure 3: Code example for using uncertainty heads.

6 Collection of Pre-trained Uncertainty Heads for Popular LLMs

We pre-trained a collection of UQ heads for a range of popular 7B–9B LLMs, including Mistral, LLaMA series, and Gemma 2. In addition to model-level UQ, we release token-level UQ heads that can provide uncertainty scores directly for tokens without explicit claim annotation, which enables broader applicability. Our UQ heads are designed for use as an off-the-shelf tool for confidence estimation in LLMs. They could be loaded from the hub using a procedure similar to the "from_pretrained" API in the HuggingFace Transformers library and integrated into the LLM generation procedure with an adapter. A code example is provided in Figure 3. Thus, UQ heads could be integrated into third-party code with minimal modifications and could be used as a plug-and-play solution for researchers and practitioners. Examples of UQ head predictions are in Appendix G.

7 Conclusion and Future Work

We presented pre-trained UQ heads – supplementary supervised modules for LLMs that help to capture their uncertainty much more effectively than unsupervised UQ methods. We demonstrated that they are quite robust and deliver state-of-the-art results for both in-domain and out-of-domain prompts. They also show remarkable generalization to other languages. Inspired by their good performance, we pre-trained a collection of UQ heads for a series of popular LLMs, including Mistral, Gemma 2, and LLama series. We release the code and the pre-trained uncertainty heads so they could be used as off-the-shelf hallucination detectors for other researchers and practitioners. In future work, we plan to scale up the training data and explore the limits of the supervised approach to UQ.

Limitations

Uncertainty heads cannot solve the problem when LLMs are trained to provide misinformation. In this situation, models are confident in their deceptive answers. Uncertainty heads cannot provide ideal annotation of hallucinations, as some LLMs do not have enough capacity to provide information about what they know and what they do not know. While we see generalization in uncertainty heads, we should acknowledge that, as with any other supervised method, they work best for "in-domain" data. The bias present in LLMs could also be transferred into uncertainty heads.

Ethical Considerations

In our work, we considered open-weight 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 a more reliable use of neural networks.

Despite that our proposed method demonstrates sizable performance improvements, it can still mistakenly highlight correctly generated text with high uncertainty in some cases. Thus, as with other uncertainty quantification methods, it has limited applicability and users should be aware of the limitations of this technology.

We release our source code under the MIT license for broader adoption. We use writing assistants to ensure grammatical correctness throughout the text.

References

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*.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4401–4420, Bangkok, Thailand. Association for Computational Linguistics.

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.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. *Preprint*, arXiv:2407.07071.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. 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, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9367–9385, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Jianfeng He, Linlin Yu, Shuo Lei, Chang-Tien Lu, and Feng Chen. 2024a. 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.

Jinwen He, Yujia Gong, Zijin Lin, Yue Zhao, Kai Chen, et al. 2024b. Llm factoscope: Uncovering llms' factual discernment through measuring inner states. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10218–10230.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Yukun Huang, Yixin Liu, Raghuveer Thirukovalluru, Arman Cohan, and Bhuwan Dhingra. 2024. Calibrating long-form generations from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13441–13460, Miami, Florida, USA. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b. *CoRR*, abs/2310.06825.

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. 2022. Language models (mostly) know what they know. *CoRR*, abs/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.

Qing Li, Jiahui Geng, Chenyang Lyu, Derui Zhu, Maxim Panov, and Fakhri Karray. 2024. Reference-free hallucination detection for large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4542–4551, Miami, Florida, USA. Association for Computational Linguistics.

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.

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.

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.

Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems*, volume 37, pages 134507–134533. Curran Associates, Inc.

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*.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625, Singapore. Association for Computational Linguistics.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Lyudmila Rvanova, Sergey Petrakov, Alexander Panchenko, et al. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *arXiv preprint arXiv:2406.15627*.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252, Dublin, Ireland. 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.

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.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. 2023. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*.

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. 2024a. LUQ: Long-text uncertainty quantification for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5244–5262, Miami, Florida, USA. Association for Computational Linguistics.

Caiqi Zhang, Ruihan Yang, Zhisong Zhang, Xinting Huang, Sen Yang, Dong Yu, and Nigel Collier. 2024b. Atomic calibration of llms in long-form generations. *Preprint*, arXiv:2410.13246.

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.

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 Training Data Generation Pipeline

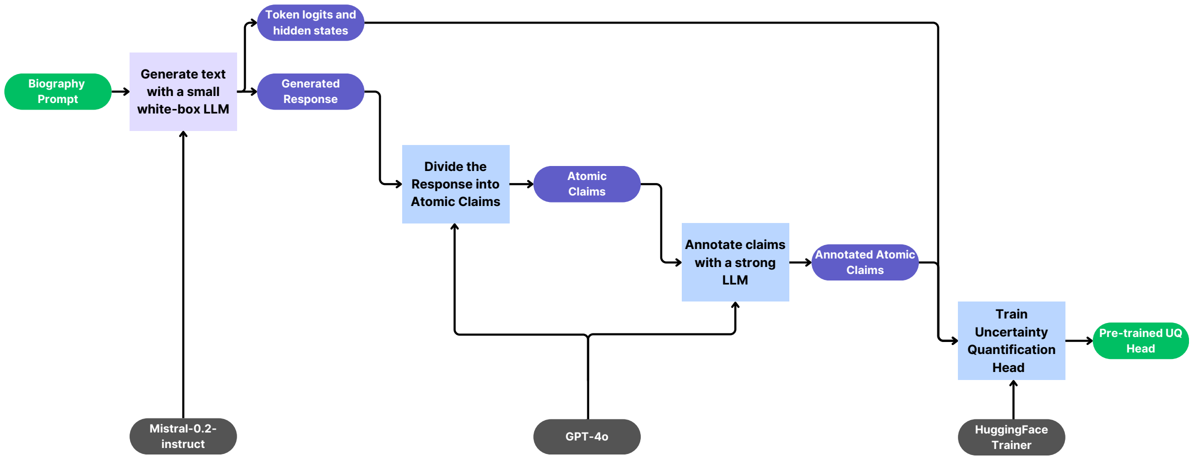

Token logits and hidden states
Biography Prompt
Generate text with a small white-box LLM
Generated Response
Divide the Response into Atomic Claims
Atomic Claims
Annotate claims with a strong LLM
Annotated Atomic Claims
Train Uncertainty Quantification Head
Pre-trained UQ Head
Mistral-0.2-instruct
GPT-4o
HuggingFace Trainer

Figure 4: The training data generation pipeline.

B Dataset Details

B.1 Dataset Construction

We used few-shot learning to better guide the LLM to generate the items for the desired domain. The structure of the prompts looks as follows:

```
Continue the list of 100 most famous {domain items}:

1. <domain-item-1>

2. <domain-item-2>

3. <domain-item-3>
```

Example for the “cities” domain:

```
Continue the list of 100 most famous cities:

1. Paris, France

2. Amsterdam, Netherlands

3. Osaka, Japan
```

For claim extraction and their annotation, we use GPT-4o with prompts from (Fadeeva et al., 2024). Overall expenses for LLM API calls are approximately $4000.

B.2 Dataset Statistics

Table 4 presents the statistics of the datasets used for training and validation; Table 5 shows the statistics of the datasets used for testing.

Model	Dataset	# of texts	# of claims	Claim accuracy, %
Mistral 7b Instruct v0.2	biographies	3,300	68,241	73.7
	multi-domain	700	14,554	86.0
Gemma 2 9b Instruct	biographies	3,300	83,716	88.6

Table 4: Statistics about the training datasets used in our experiments.

Split	# of prompts	ChatGPT prompt used to generate questions	Testing prompt	# of claims		Claim accuracy, %	
				Mistral	Gemma	Mistral	Gemma
persons	100	Tell me a list of 100 most famous persons.	Tell me a bio of a <person>	2,234	2,857	72.9	87.4
cities	100	Tell me a list of 100 most famous cities.	Tell me a history of a <city>	2,128	2,684	79.8	87.1
movies	100	Tell me a list of 100 most famous movies.	Tell me about the movie <movie> and its cast.	2,568	3,121	89.7	94.8
inventions	100	Tell me a list of 100 most important inventions.	Tell me about the invention of <invention> and its inventor.	2,269	2,626	84.3	92.1
books	100	Tell me a list of 100 most famous books.	Tell me about the book <book> and its author.	2,530	3,070	89.9	95.9
artworks	100	Tell me a list of 100 most famous artworks.	Tell me about the artwork <artwork> and its artist.	2,464	2,873	75.9	85.1
landmarks	100	Tell me a list of 100 most famous landmarks.	Tell me about the landmark <landmark>.	2,365	2,566	88.5	93.7
events	100	Tell me a list of 100 most significant historical events.	Tell me about <event> event.	2,294	2,665	88.9	94.8
Russian	100	—	Расскажи биографию <person>	—	3,572	—	66.7
Chinese	100	—	介绍一下<person>	—	2,248	—	77.8
German	100	—	Erzhlen Sie mir eine Biografie von <person>	—	2,815	—	85.1

Table 5: The statistics of the multi-domain test dataset and number of claims generated my Mistral 7B Instruct v0.2 and Gemma 2 9b Instruct models.

C Hardware and Computational Efficiency

All experiments were conducted on 8 NVIDIA RTX 5880 Ada GPUs. On average, training a single model with hyperparameter search takes around 150 GPU hours.

Method	Computational Overhead	GPU Memory Footprint
MCP	0.0 %	-
Perplexity	0.0 %	-
Max Token Entropy	0.2 %	-
CCP	8.6 %	1,546 MB
SAPLMA	4.7 %	4 MB
Factoscope	6.1 %	32 MB
Lookback Lens	5.5 %	<1 MB
UHead (only hidden states)	8.7 %	73 MB
UHead (att. + prob. + hs.)	9.9 %	82 MB
UHead (Ours)	4.9 %	40 MB

Table 6: Computational overhead of UQ methods evaluated with the Mistral 7B Instruct v0.2 model. Overhead is measured relative to the fastest method, MCP. For CCP, the size of the auxiliary NLI model is reported. The results were obtained using a multi-domain dataset containing 800 texts and a total of 18,852 claims.

D Analysis of Attention-Based Features

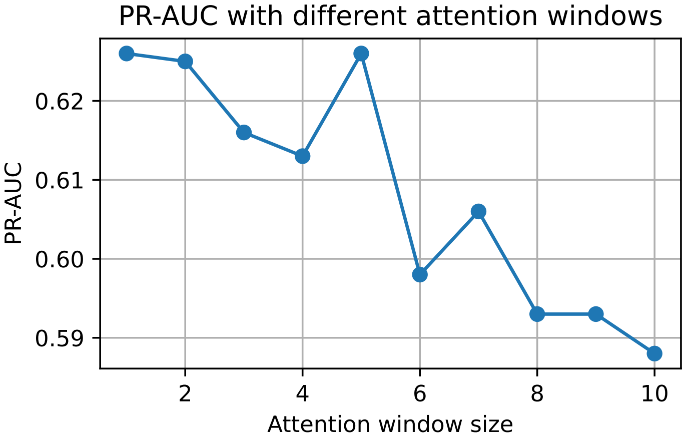


Figure 5: PR-AUC for different attention window sizes using UHead for the Mistral 7B Instruct v0.2 model.

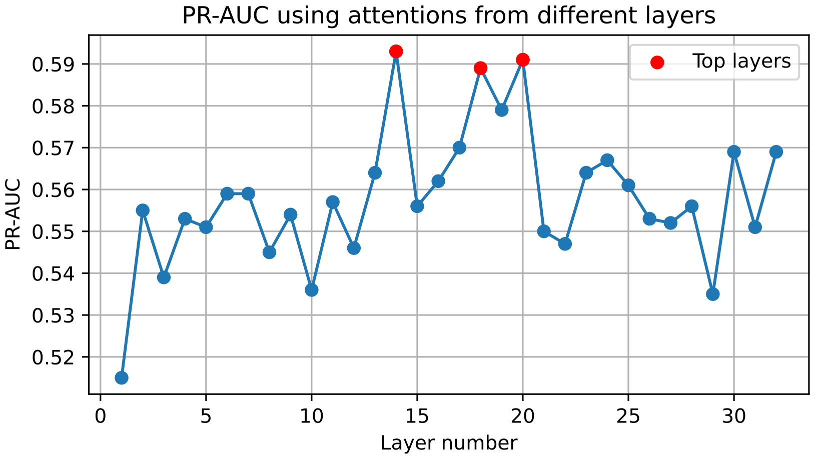


Figure 6: PR-AUC as a function of layer number used for attention features in UHead for the Mistral 7B Instruct v0.2 model. Highlighted points mark layers with highest PR-AUC (layers 14, 18 and 20).

E Additional Experimental Results

Method	**PR-AUC**
MCP	.165
Perplexity	.234
Max Token Entropy	.306
CCP	.363
UHead	**.424**

Table 7: PR-AUC for various UQ methods on 200 questions from the TruthfulQA dataset. The results show generalization of UHead trained on *biographies* to the QA task.

Architecture	**PR-AUC**
Linear	.556
MLP	.626
Transformer (UHead, ours)	**.642**

Table 8: PR-AUC for different UQ head architectures for the Mistral 7B Instruct v0.2 model on the dev set of *biographies* dataset. The hyperparameters of all detectors are optimized. The results demonstrate the superiority of the Transformer architecture.

Method	**PR-AUC**
MCP	.180
CCP	.307
UHead trained on native dataset (Gemma)	**.461**
UHead trained on non-native dataset (Mistral)	.315

Table 9: PR-AUC of the hallucination detector for Gemma 2 trained on the “native” data (generated by Gemma 2) in comparison to training on “non-native” data (generated by Mistral). PR-AUC is reported on the test set of English *biographies* dataset. The results show that using “non-native” training data substantially reduces the performance

Test Sets **Method**	**Cities**	**Movies**	**Inventions**	**Books**	**Artworks**	**Landmarks**	**Events**
UHead, bio	.487	.466	.485	.395	.561	.340	.369
UHead, bio + all - 1	.489	.479	.482	.404	.572	.338	.387

Table 10: Introducing more diverse training data. UHead results are shown for two scenarios: when the UQ head is trained solely on the English biographies dataset, and when it is trained on the biographies dataset along with all other domains, excluding the test domain. Adding more data slightly improves the performance in the OOD setting.

F Hyperparameters

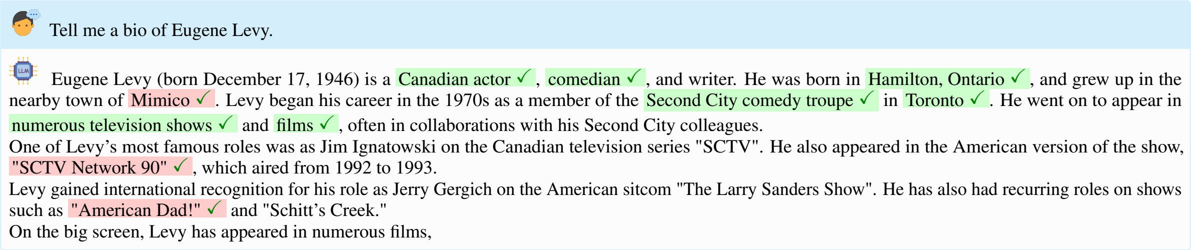
Method	Model	Learning Rate	Num. Epochs	Weight Decay	Dropout Rate	Warmup	Att. Window Size	Transformer arch.
SAPLMA	Gemma 2 9b Instruct	1e-4	10	0.1	0.1	0.1	–	–
	Mistral 7b Instruct v0.2	1e-4	10	0.1	0.1	0.1	–	–
Lookback Lens	Gemma 2 9b Instruct	1e-2	13	0.1	0.1	0.1	–	–
	Mistral 7b Instruct v0.2	1e-2	13	0.1	0.1	0.1	–	–
UHead (Factoscope)	Gemma 2 9b Instruct	2e-4	3	0.1	0.2	0.05	–	1 layer / 768 width / 16 heads
	Mistral 7b Instruct v0.2	2e-4	5	0.2	0.2	0.05	–	1 layer / 256 width / 4 heads
UHead	Gemma 2 9b Instruct	2e-4	6	0.1	0.05	0.1	2	1 layer / 768 width / 16 heads
	Mistral 7b Instruct v0.2	2e-4	7	0.1	0.2	0.1	2	2 layers / 768 width / 4 heads

Table 11: Optimal hyperparameters for each method and model.

For each tested model, we selected hyperparameters by optimizing the PR-AUC metric on the validation set of the “biographies” dataset. In training, we optimized the learning rate, warmup ratio, number of epochs, and the weight of positive examples in the cross-entropy loss. For the model architecture, we optimized the number of uncertainty layers, the number of heads, and the intermediate dimension. For feature extraction, we optimized the number of layers used to obtain hidden states, token probabilities, and attention weights, as well as the number of preceding tokens considered for attention. The optimal hyperparameters are summarized in Table 11. The hyperparameter grid is the following:

Learning rate: {1e-5, 3e-5, 5e-5, 1e-4, 2e-4, 5e-4, 1e-2};
Num. of epochs: $\{n \in \mathbb{N} \mid 2 \leq n \leq 15\}$;
Warmup: {0., 0.05, 0.1};
Attention window size: {1, 2, 3, 4, 5, 10};
Dropout rate: {0., 0.05, 0.1, 0.2};
Weight decay: {0, 1e-2, 1e-1}.

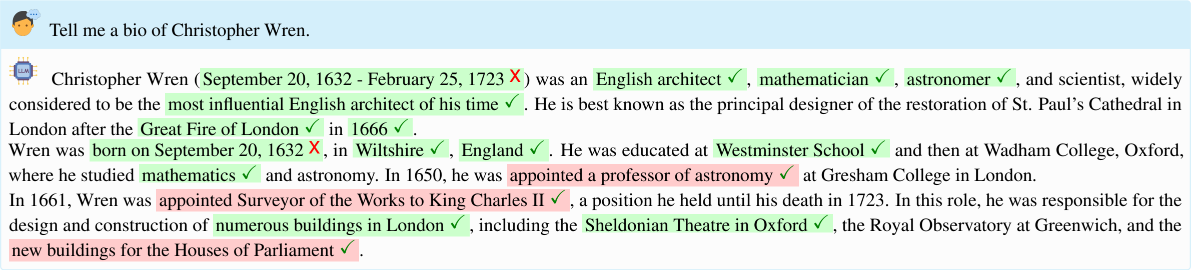
G Examples of Hallucinations and UQ Head Predictions



Tell me a bio of Eugene Levy.

Eugene Levy (born December 17, 1946) is a Canadian actor ✓ , comedian ✓ , and writer. He was born in Hamilton, Ontario ✓ , and grew up in the nearby town of Mimico ✓ . Levy began his career in the 1970s as a member of the Second City comedy troupe ✓ in Toronto ✓ . He went on to appear in numerous television shows ✓ and films ✓ , often in collaborations with his Second City colleagues.
One of Levy's most famous roles was as Jim Ignatowski on the Canadian television series "SCTV". He also appeared in the American version of the show, "SCTV Network 90" ✓ , which aired from 1992 to 1993.
Levy gained international recognition for his role as Jerry Gergich on the American sitcom "The Larry Sanders Show". He has also had recurring roles on shows such as "American Dad!" ✓ and "Schitt's Creek."
On the big screen, Levy has appeared in numerous films,

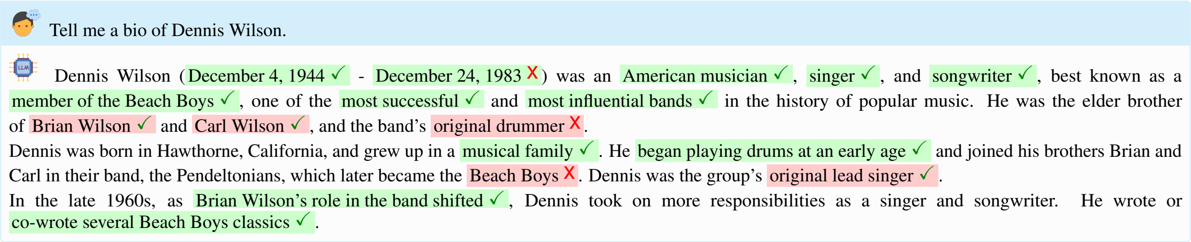
Figure 7: Mistral's response to the question *"Tell me a bio of Eugene Levy"* and UHead predictions. Claim-relevant phrases are highlighted only when the model shows high confidence (green, uncertainty < 0.2) or high uncertainty (red, uncertainty ≥ 0.6). Medium-uncertainty phrases (0.2–0.6) are left unmarked. Ticks (✓) indicate confidence matches ground-truth label; crosses (x) indicate mismatch.



Tell me a bio of Christopher Wren.

Christopher Wren (September 20, 1632 - February 25, 1723 X) was an English architect ✓ , mathematician ✓ , astronomer ✓ , and scientist, widely considered to be the most influential English architect of his time ✓ . He is best known as the principal designer of the restoration of St. Paul's Cathedral in London after the Great Fire of London ✓ in 1666 ✓ .
Wren was born on September 20, 1632 X , in Wiltshire ✓ , England ✓ . He was educated at Westminster School ✓ and then at Wadham College, Oxford, where he studied mathematics ✓ and astronomy. In 1650, he was appointed a professor of astronomy ✓ at Gresham College in London.
In 1661, Wren was appointed Surveyor of the Works to King Charles II ✓ , a position he held until his death in 1723. In this role, he was responsible for the design and construction of numerous buildings in London ✓ , including the Sheldonian Theatre in Oxford ✓ , the Royal Observatory at Greenwich, and the new buildings for the Houses of Parliament ✓ .

Figure 8: Mistral's response to the question *"Tell me a bio of Christopher Wren"* and UHead prediction. Claim-relevant phrases are highlighted only when the model shows high confidence (green, uncertainty < 0.2) or high uncertainty (red, uncertainty ≥ 0.6). Medium-uncertainty phrases (0.2–0.6) are left unmarked. Ticks (✓) indicate confidence matches ground-truth label; crosses (x) indicate mismatch.



Tell me a bio of Dennis Wilson.

Dennis Wilson (December 4, 1944 ✓ - December 24, 1983 X) was an American musician ✓ , singer ✓ , and songwriter ✓ , best known as a member of the Beach Boys ✓ , one of the most successful ✓ and most influential bands ✓ in the history of popular music. He was the elder brother of Brian Wilson ✓ and Carl Wilson ✓ , and the band's original drummer X .
Dennis was born in Hawthorne, California, and grew up in a musical family ✓ . He began playing drums at an early age ✓ and joined his brothers Brian and Carl in their band, the Pendeltonians, which later became the Beach Boys X . Dennis was the group's original lead singer ✓ .
In the late 1960s, as Brian Wilson's role in the band shifted ✓ , Dennis took on more responsibilities as a singer and songwriter. He wrote or co-wrote several Beach Boys classics ✓ .

Figure 9: Mistral's response to the question *"Tell me a bio of Dennis Wilson"* and UHead predictions. Claim-relevant phrases are highlighted only when the model shows high confidence (green, uncertainty < 0.2) or high uncertainty (red, uncertainty ≥ 0.6). Medium-uncertainty phrases (0.2–0.6) are left unmarked. Ticks (✓) indicate confidence matches ground-truth label; crosses (x) indicate mismatch.