Confidence-Based Response Abstinence: Improving LLM Trustworthiness via Activation-Based Uncertainty Estimation

Zhiqi Huang[†], **Vivek Datla**[†‡], **Chenyang Zhu, Alfy Samuel,**
Daben Liu, **Anoop Kumar**, **Ritesh Soni**
{zhiqi.huang, vivek.datla, chenyang.zhu, alfy.samuel,
daben.liu, anoop.kumar, ritesh.soni}@capitalone.com
[†]Equal Contribution, [‡]Corresponding Author

Abstract

We propose a method for confidence estimation in retrieval-augmented generation (RAG) systems that aligns closely with the correctness of large language model (LLM) outputs. Confidence estimation is especially critical in high-stakes domains such as finance and healthcare, where the cost of an incorrect answer outweighs that of not answering the question. Our approach extends prior uncertainty quantification methods by leveraging raw feed-forward network (FFN) activations as auto-regressive signals, avoiding the information loss inherent in token logits and probabilities after projection and softmax normalization. We model confidence prediction as a sequence classification task, and regularize training with a Huber loss term to improve robustness against noisy supervision. Applied in a real-world financial industry customer-support setting with complex knowledge bases, our method outperforms strong baselines and maintains high accuracy under strict latency constraints. Experiments on Llama 3.1 8B model show that using activations from only the 16th layer preserves accuracy while reducing response latency. Our results demonstrate that activation-based confidence modeling offers a scalable, architecture-aware path toward trustworthy RAG deployment.

1 Introduction

In high-stakes applications like financial customer support, it is often more desirable and trustworthy for a Retrieval Augmented Generation (RAG) system to abstain from answering than to risk providing an incorrect response. Although not responding to a query reduces the system's immediate utility, it is a necessary trade-off to ensure accuracy and preserve user trust. The guiding principle is that the reputational and financial cost of providing a wrong answer is significantly higher than the cost of not providing one. This challenge requires a principle of abstention.

One way to achieve the abstention is to have a confidence measure that correlates with correctness of the response, and mask the response when the confidence score is below a threshold. Uncertainty of the model while generating the response is a viable source of signal for building a confidence measure.

To develop a practical solution, it is crucial to identify the primary source of this uncertainty. In highly regulated fields, the error is rarely due to aleatoric uncertainty (randomness inherent in the data), as knowledge bases are typically vetted by legal and subject-matter experts. The more probable source is epistemic uncertainty (the model's own lack of knowledge), which arises when the model's parametric knowledge, acquired during pre-training or fine-tuning, conflicts with or misinterprets the provided context.

While existing approaches (Bakman et al., 2024; Liu et al., 2024; Malinin and Gales, 2020; Kuhn et al., 2023) to uncertainty estimation in retrieval-augmented generation (RAG) have shown promise, they often fall short when the target response is long and narrative in nature. This challenge becomes especially pronounced in sensitive domains such as finance, where queries can be ambiguous or underspecified. For instance, a question like "What is the deadline to make a payment on Card Type A?" may retrieve multiple similar documents, each corresponding to different subcategories of the card type. In such cases, both the query and the retrieved context exhibit ambiguity, which can propagate through the RAG pipeline. Simply measuring uncertainty based on generated response is insufficient to ensure correctness.

Also, methods relying on sampling (Bakman et al., 2024), are less practical at scale. These techniques rely on generating a response multiple times with slight variations to measure the model's consistency, a process that introduces prohibitive computational costs and latency in a production en-

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 184–193

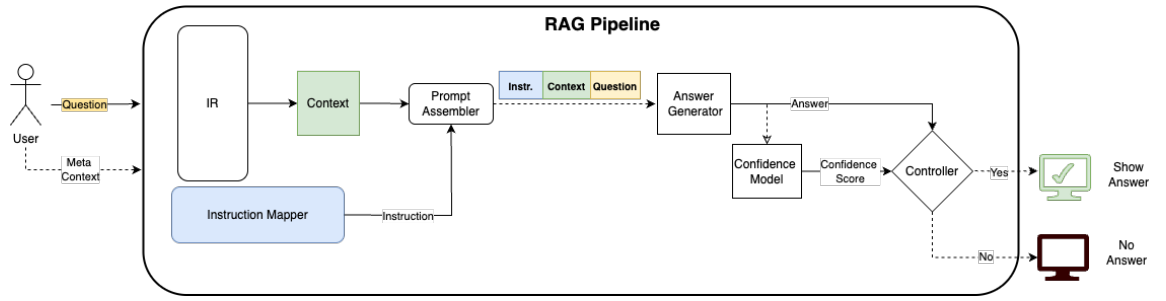

Figure 1: Diagram of the proposed Retrieval Augmented Generation (RAG) with the confidence model. When a user asked a question, the IR component retrieves related context from a database. The prompt is then constructed and sent into a question and answering LLM. A confidence score would be generated by the confidence model and being used to control whether or not to show the result to the user.

vironment. For RAG systems that must serve users in real-time, such multi-generational approaches are not a viable solution.

Uncertainty and correctness, while related, are fundamentally distinct concepts (Liu et al., 2025). A model's low uncertainty in its output does not necessarily imply correctness, just as a model may generate a correct response with a high uncertainty. This distinction becomes particularly salient in retrieval-augmented generation (RAG) applications, where correctness often hinges on factual grounding rather than surface-level fluency. Our goal is to utilize the model's internal uncertainty signals to generate a confidence score that correlates strongly with the correctness of the response generated by an LLM.

We build our confidence model using the raw activation signals inside the feedforward layers of LLM which include the activations of knowledge neurons (Azaria and Mitchell, 2023). Thus, our model captures the relationship between the autoregressive properties of activations and inherent uncertainty of the model in generating a response. We propose a supervised framework to train a sequence classifier model and generate a confidence score that correlates with response correctness.

Figure 1 illustrates the practical utility of integrating a confidence model into our RAG pipeline. The primary goal of the system is to provide users with accurate answers. However, in cases where there is insufficient epistemic or aleatoric knowledge to reliably answer a question, the system's next best action is to abstain from answering. This behavior is enabled by a controller that filters responses based on their confidence scores, allowing the system to avoid potentially incorrect or misleading outputs. This system is deployed in production for large-scale use that achieves high precision while maintaining an acceptable display rate (defined as the ratio of response pass the confidence filter to total responses generated by the system). Experimental results show that our confidence model outperforms multiple baselines, reaching a precision of 0.95 with 70.1% display rate (masking 29.9% of the total responses). Furthermore, when compared to ground truth, displayed responses exhibit a significantly higher ROUGE score than masked responses.

2 Related Work

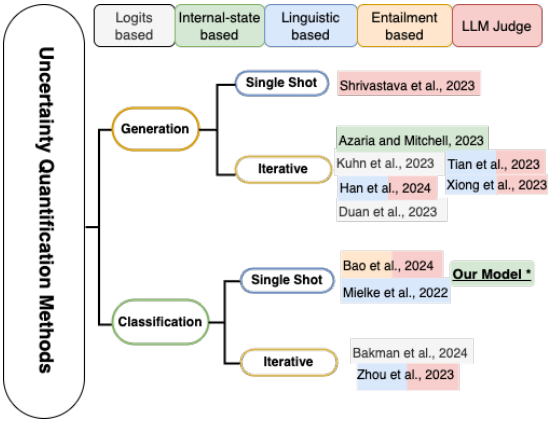

Figure 2: Landscape of Uncertainty Quantification Methods

Figure 2 shows the landscape of various uncertainty quantification methods in LLMs. When mapping the landscape, they can be broadly grouped by the strategies used to quantify the uncertainty.

Shrivastava et al. (2023) demonstrate that the generation probabilities of weaker white-box models (that is, smaller models) can be used to estimate the internal confidence levels of larger black-box models. The approach involves zero-shot generation using prompt variations based on dif-

ferent instructions to infer the confidence of responses produced by the larger model. Duan et al. (2023) and Kuhn et al. (2023) use semantic entropy to reweight token-level importance, prioritizing content-bearing tokens while discounting filler words. Their core intuition is that if semantically important tokens are generated with high confidence, the overall response is more likely to be correct even if less important tokens exhibit lower confidence.

Azaria and Mitchell (2023) show that the LLMs internal parameters show tell-tale signs when generating text with uncertainty. When a model's generation path falls into a speculative region, evidenced by competition between two or more plausible next tokens, its confidence is adversely affected. They introduce small input perturbations to induce trajectory shifts, and monitor corresponding changes in token-generation activations and outputs. They label the speculative generation as lying, and propose that activation patterns can shed light on this speculative generation. This method requires white-box access to the model to obtain token-level probability traces.

Tian et al. (2023) have empirically shown that LLM's self generated confidence score while giving a response could be calibrated by sampling over perturbed questions. Specifically, they show that prompting a model to produce several answer choices before giving its confidence scores helps in calibration of verbalized probabilities.

In a related direction, Xiong et al. (2023) generate multiple variants of a prompt using diverse prompting strategies such as Chain-of-Thought (CoT), self-probing, and top-k sampling. They utilize a separate LLM as a "judge model" to evaluate each variant and assign a confidence score. Variations in these scores are then used to predict the confidence of the target model's original response. Similarly, Han et al. (2024) proposed a confidence measurement based on the perturbation of the question. The variation in model's answer generation probabilities for various perturbations of the question for the same context is used as a measure to generate a verbalized confidence score.

Several recent studies adopt a classification-based approach to estimate response plausibility, offering a more computationally efficient alternative by avoiding multiple generations. For example, HHEM (Bao et al., 2024) uses an entailment-based model to measure the semantic coherence between the input and the generated output. This approach operates under black-box constraints, requiring only the input-output pair from the target LLM to assess the correctness of the response.

Other methods focus on linguistic cues as indicators of ambiguity in LLM outputs. Mielke et al. (2022) argue that model confidence does not always correlate with correctness and show that linguistic calibration of input prompts can significantly influence a model's confidence. They introduce a calibration score that helps generate more accurate responses by aligning linguistic features with expected confidence levels. Their evaluations were performed on factoid QA datasets, where there is a zero-sum approach towards correctness. We argue that when the parametric knowledge of the LLM is mainly contributing to the style of the response, and the key facts come from the input, confidence can serve as an effective signal for correctness.

Our method draws inspiration from prior work on activation-based knowledge tracing (Dai et al., 2022), generation trajectory modeling (Azaria and Mitchell, 2023), and importance-weighted token probabilities (Bakman et al., 2024). Dai et al. (2022) highlight how feedforward network (FFN) activations encode key factual information, showing that the activation of certain neurons is positively correlated with knowledge expression. Building on this insight, we treat FFN activations as autoregressive signals and train a recurrent neural network (RNN) to predict the probability that a model-generated response is correct. A score closer to 1 indicates greater model confidence in the response's correctness.

3 Method

For a generated response sequence s of length L for the given input x to a model M with parameters θ, the probability of generating the sequence is given as follows:

$$P(s \mid x; \theta) = \prod_{l=1}^{L} P(s_l \mid s_{<l}, x; \theta) \quad (1)$$

To compare sequence probability across different lengths of generated output, previous approaches have normalized the score based on the length of the response. The length-normalized score, used in prior uncertainty estimation (UE) methods (Malinin and Gales, 2020):

$$\tilde{P}(s \mid x; \theta) = \left(\prod_{l=1}^{L} P(s_l \mid s_{<l}, x; \theta) \right)^{1/L} \quad (2)$$

Here all the tokens contribute are given equal importance irrespective of the length of sequence. The risk with this approach is that a single low-probability unusual word can disproportionately lower the overall sequence score, even if subsequent tokens have high probabilities.

Several of the methods that perform uncertainty estimation taking token-logits perform similar weighing and they have shown great results in factoid question answering. These methods do not scale for longer answers, where there are multiple sentences and few tokens don't hold the key to correctness. Also, multiple generations needed to quantify the confidence score make them prohibitively expensive in a large scale settings.

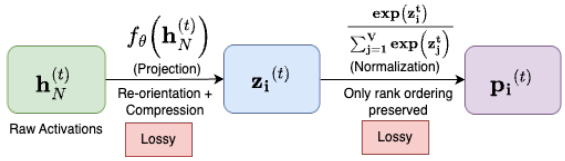

Figure 3: Motivation to use activations instead of token probabilities.

Our goal is to estimate the correctness of the generated response in a single shot using uncertainty estimation. We prefer using FF-layer activations rather than token probabilities because token probabilities are computed by applying the decoder head (a linear projection) followed by a softmax transformation. This projection compresses the rich internal representation into a vocabulary space and the softmax operation further distorts the signal by normalizing it into a probability distribution (see Figure 3), potentially obscuring fine-grained differences in the model's internal state. In contrast, raw activations preserve the high-dimensional representation prior to this compression, providing a more direct view of the model's internal dynamics during response generation.

3.1 Our Confidence Model

Figure 4 shows a graphical representation of our confidence model. To estimate the confidence of a generated answer s of size L, we introduce a lightweight, trainable probe that operates on the internal representations of the Llama 3.1 8B model. The process begins by providing a structured prompt to the LLM, which is formulated as a sequence of tokens, x of size $T + L + 1$, which is a concatenation of the following: Instruction(x_I), Question(x_Q) and Context(x_C) of size T tokens; Answer(s) of size L tokens; and EOS token(x_{EOS}) of size 1. The complete input sequence is formally represented as:

$$x = x_I \oplus x_Q \oplus x_C \oplus s \oplus x_{EOS} \tag{3}$$

where $\oplus$ denotes the concatenation operation.

During a single forward pass through the LLM, we extract the hidden state activations from a specific transformer layer, ℓ. We investigate representations from two distinct depths within the network: the final layer ($\ell = 32$) and a middle layer ($\ell = 16$). This yields a full sequence of hidden state vectors

$$\mathbf{H}_\ell = (\mathbf{h}_\ell^1, \ldots, \mathbf{h}_\ell^{T+L+1}) \tag{4}$$

Each vector $\mathbf{h}_{\ell,k} \in \mathbb{R}^{d_{\text{LLM}}}$ corresponds to the k-th input token, with size of LLM's activation dimension. From this complete set of activations, we isolate only those corresponding to the tokens of the candidate answer, which span from index $T+1$ towards the final x_{EOS} token. This forms the input sequence, S_{in}, for our confidence estimation module:

$$S_{\text{in}} = (\mathbf{h}_\ell^{T+1}, \mathbf{h}_\ell^{T+2}, \ldots, \mathbf{h}_\ell^{T+L+1}) \tag{5}$$

The extracted sequence S_{in} is then fed into a sequence classifier $g(S_{in})$, which is trained to model the sequence of activations. The sequence classifier with a classification head outputs a 2-dimensional logit vector, $\mathbf{z}$, such that the confidence score can be computed as,

$$c = \text{softmax}(\mathbf{z})_1 = \frac{e^{z_1}}{e^{z_0} + e^{z_1}} \tag{6}$$

Our goal is to estimate the confidence of the model when generating an answer, with ulterior goal of rejecting the generated answer if c falls below a threshold of confidence. In this framework, only the parameters of the sequence classifier $g(S_{in})$ are trainable. We use a Long short-term memory (LSTM) (Sutskever et al., 2014) as the sequence classifier for the following experiments.

3.2 Model Training

Given that the retrieval stage of the pipeline may introduce alethic knowledge gaps, the input context provided to the LLM can be incomplete, or contain contradictory information across the document retrieved. To address this, we introduce an explicit regularizer based on Huber loss L_{Huber}, which is more robust to such noise (Patra et al., 2023). Unlike just using only the Cross-Entropy loss L_{CE}, which can be highly sensitive to large deviations

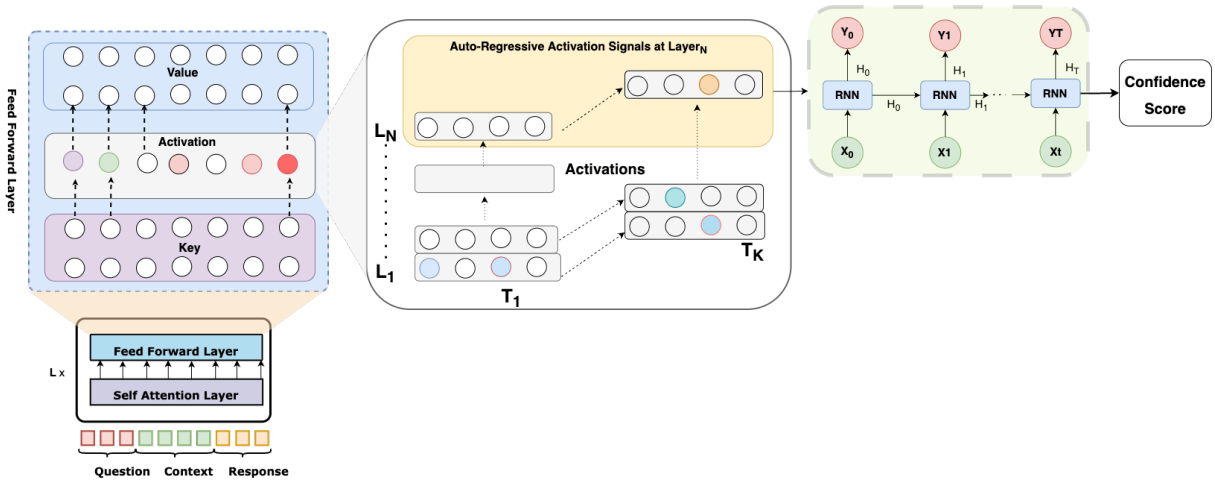

Figure 4: Confidence model based on the activations of large language models. Our method first feed the <Question, Context, Response> pair in an LLM. We then extract the activations from the 32-th or 16-th layer, and feed the activations into an LSTM and a classification head. The classification logit serves as the confidence score.

when predictions are far from the target, the Huber loss based regularizer helps smoothen with a linear penalty for large errors. This property reduces the influence by outliers arising from imperfect retrieval.

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \le \delta \\ \delta\left(|x| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \quad (7)$$

where $\delta > 0$ is a hyperparameter that controls the transition point between the quadratic and linear loss.

Using L_{CE} loss with L_{Huber} regularizer, we learn to predict confidence score, which correlates with the correctness. The higher the confidence, the higher are the changes for the generated output to be correct. For a sampled minibatch $B = \{(x_j, y_j)\}_{j=1}^{|B|}$, the Huber loss term is calculated as:

$$L_{\text{Huber}} = H_\delta\left(\frac{1}{|B|}\sum_{i=1}^{|B|} c_i - \frac{1}{|B|}\sum_{i=1}^{|B|} I(\hat{y}_i = y_i)\right) \quad (8)$$

where $c_i = \max(\hat{y}_i)$ is the confidence of the prediction for instance x_i, and $I(\hat{y}_i = y_i)$ is the indicator function for correct predictions.

The total loss function

$$L_{Total} = L_{CE} + \lambda L_{Huber} \quad (9)$$

where λ controls the strength of regularization.

In our modeling, several constraints arise naturally from the real-time conditions under which the system operates. The generated output must remain grounded in the input context provided within the prompt. The output must adhere to predefined stylistic or structural patterns required to present certain types of information. At the end of generation, an explicit decision signal determines whether the answer is shown to the user. This signal is conditioned on multiple factors, including:

- Subject-matter-expert (SME) defined standards of correctness for the class of questions.
- The requirement that factual content be derived from the input context, while stylistic elements may rely on the model's parametric knowledge.

We conducted experiments on our proprietary knowledge corpus consisting of procedures, rules, and complex instructions to be followed to address the various needs of support agents handling a large volume of customer base. Our results indicate a robust performance using our method compared to the several SOTA UQ and hallucination identification methods.

4 Experimentation

We have conducted experiments to identify the optimal masking ratio in order to maintain utility and precision of the system.

4.1 Data

4.1.1 Disclosure on data

Due to the sensitive nature of the data, which pertains to proprietary financial tools and internal knowledge resources used by service agents

within a financial institution, we are unable to share dataset details. This restriction is in place to ensure compliance with internal data governance policies and to protect confidential and regulated financial information. We hope that the community understands the importance of maintaining the integrity and privacy of such sensitive operational data.

4.1.2 Features of our knowledge articles

We provide an overview of the population-level characteristics of our dataset, which is derived from a knowledge base composed of instructional articles designed to guide customer support agents in using proprietary internal tools. These tools are governed by strict procedural guidelines essential for resolving customer issues. For instance, when handling a customer inquiry about a specific transaction, agents must follow a prescribed sequence: verifying the customer's identity, obtaining consent to access the account, identifying the relevant transaction, and initiating additional processes such as flagging the transaction in cases of suspected fraud.

Figure 5(a) shows the hierarchical nature of the documents. Our knowledge base is semi-structured comprising heterogeneous documents with rich hierarchical and content structures. These documents may include deeply nested sections (e.g., sections, subsections, sub-subsections), as well as complex content types such as tables, bullet and numbered lists, and embedded entities.

Each subsection article is treated as a separate document. Each document is further chunked to be efficiently indexed in a low latency store. Overall, there are $8.5k$ documents and approximately $45k$ chunks in the knowledge-base.

4.1.3 Features of the training data

Our system design incorporates a real-time feedback loop, as illustrated in Figure 5(b), where support agents interact with the RAG system and provide immediate feedback (e.g., thumbs-up/down) on the usefulness of generated responses. Processing thousands of these interactions daily, we draw a stratified sample of both positive and negative feedback instances, accounting for dimensions like product type and line of business. For each sampled case, we collect the query, generated answer, retrieved context, and associated metadata for a more rigorous offline evaluation.

This offline review is conducted by subject matter experts (SMEs) who assess each answer for completeness, correctness, and truthfulness, ensuring it is grounded in the provided context rather than inferred from the model's parametric knowledge. SMEs may also refine responses to create ideal, complete answers, as shown in the example in Figure 5(c). This two-tiered approach of combining real-time user signals with deep SME validation allows us to build a high-quality labeled dataset for training and evaluation, ensuring the model aligns with domain-specific requirements for accuracy and trustworthiness.

4.2 Information Retrieval

We perform retrieval using an open-search index configured for K-nearest neighbor (KNN) retrieval based on semantic similarity to the input query. In addition to the query itself, we incorporate associated metadata such as entitlements and access-control filters specific to the agent submitting the question, to ensure that the retrieved documents adhere to the agent's permissions.

In the context of this work, we do not explicitly quantify retrieval errors. Instead, our focus lies in modeling the generation process of the response. We assume the retrieval step to be correct and treat errors introduced during retrieval as alethic uncertainty, while the knowledge encoded within the model through pretraining and fine-tuning is considered epistemic. Our confidence model is designed to map the relationship between the question, the retrieved (alethic) knowledge, the model's internal (epistemic) knowledge, and the generated response. This relationship is captured through patterns in the model's internal activations, treated as auto-regressive signals.

We observe that this mapping cannot be adequately modeled using a simple feedforward (MLP) architecture, as it fails to capture the temporal dependencies inherent in the generation process. Therefore, we adopt a recurrent architecture specifically, a lightweight Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM), trained using L_{CE} loss and L_{Huber} regularizer loss. The LSTM is trained on input sequences derived from the activations of a selected layer, along with carefully curated training data that aligns the activation patterns with response-level confidence.

4.3 Results

Our method achieves superior calibration of LLM responses, maintaining high precision with minimal utility loss. As shown in Table 1, it outperforms industry SOTA methods, Vectara

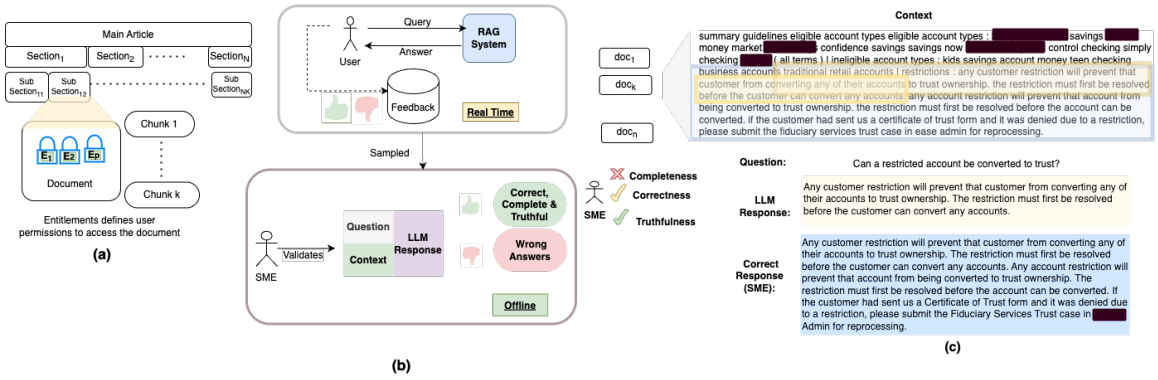

Figure 5: Features of our Knowledge base. (a) Complex structure of our knowledge articles; (b) Process of SME validated training data creation; (c) Example SME validated LLM Response

Method	AUROC
Vectara	0.590
$\text{Vectara}_{\text{FT}}$	0.634
$\text{Logits}_{\text{based}}$	0.663
$\text{Our Model}_{\text{no calib.}}$	0.741
$\text{Our Model}_{\text{with calib.}}$	**0.772**

Table 1: Comparing our approach to other baselines

Threshold	P	R	ROUGE-L Display	ROUGE-L Mask	%Mask
Baseline (0.0)	0.90	1.00	0.62	N/A	0.0
0.1	0.94	0.89	0.64	0.54	14.4
0.2	0.94	0.83	0.64	0.56	20.4
0.3	0.94	0.80	0.64	0.57	22.9
0.4	0.94	0.76	0.64	0.57	26.4
0.5	0.95	0.73	0.65	0.57	29.9
0.6	0.95	0.69	0.66	0.56	34.8
0.7	0.96	0.65	0.66	0.57	38.6
0.8	0.96	0.60	0.66	0.58	44.0
0.9	0.97	0.52	0.67	0.58	52.0

Table 2: Our Confidence score model with calibration helps achieve 0.95 precision while masking 29.9% of the responses

IR Model	R@1	R@3	R@5	R@10	R@25
	0.54	0.75	0.80	0.84	0.88

Table 3: Current recall(r) of the IR system, that helps in creating the context for the RAG pipeline

Layer	Context	P	R	ROUGE-L Display	ROUGE-L Mask	%Mask
32	Full	0.95	0.73	0.65	0.57	29.9
32	Top 5	0.95	0.69	0.66	0.56	34.3
32	Top 3	0.96	0.63	0.66	0.57	40.5
32	Top 1	0.97	0.56	0.67	0.57	47.5
16	Full	0.97	0.73	0.64	0.58	31.3
16	Top 5	0.98	0.65	0.65	0.59	39.3
16	Top 3	0.98	0.60	0.66	0.58	44.8
16	Top 1	0.99	0.48	0.66	0.59	56.2

Table 4: Identifying the optimal setting to run confidence model

(HHEM2.1) (Bao et al., 2024) and a logits-based uncertainty model (Malinin and Gales, 2020). We obtain further performance gains by caliberating with L_{Huber} as a regularizer.

Table 2 reports confidence thresholds that optimize precision while keeping the masking rate low. Although an ideal mask rate is 0%, realistic applications must tolerate some masking due to noise in LLM inputs. In our setup, the retrieval stage achieves a strong $recall@10 > 0.8$ (Table 3), yet residual alethic knowledge gaps in retrieval can affect downstream generation.

We experimented with varying input context sizes, selecting the top k documents ($k \in \{1, 3, 5, 7 \text{ (full)}\}$), and with partial-layer activation extraction from Llama 3.1 8B (layer 16 or layer 32) (AI@Meta, 2024). As shown in Table 4, using activations from only the 16th layer yields performance on par with the full-layer setup while maintaining a reasonable mask rate.

Latency analysis (Table 5) confirms that input context size is a dominant factor; larger contexts increase response time, highlighting a trade-off between context size and system responsiveness. In the production system, the confidence model is de-

Framework	Layer	Context	Avg. ms	P99
Hugging Face	32	Full	221	387
		Top 5	179	329
		Top 3	137	286
		Top 1	100	252
	16	Full	139	278
vLLM	32	Full	206	354
		Top 5	161	304
		Top 3	125	269
		Top 1	88	241
	16	Full	127	267

Table 5: Latency of the confidence model using various context sizes, Avg. time is calculated across 3 runs of the same input.

ployed with vLLM (Kwon et al., 2023), and overall the same trend appears there as well.

5 Discussion

In this work, we present an approach for constructing a confidence score that aligns with the correctness of responses generated by large language models (LLMs). Such a measure is particularly critical in high-stakes domains such as finance and healthcare, where the cost of an incorrect response far exceeds that of withholding a response. Our method extends prior works in uncertainty quantification (UQ) (Malinin and Gales, 2020; Bao et al., 2024) by leveraging model activation patterns to predict correctness more robustly.

Figure 3 illustrates our motivation for using raw activation signals from the feed-forward network (FFN) layers as auto-regressive features, rather than token logits or probabilities. Token probabilities are obtained after a linear projection and softmax transformation. The projection step reduces dimensionality, discarding non-vocabulary-aligned features, while the softmax normalization saturates probability values, erasing scale information and compressing relative differences. Using activations directly, we retain the full representational capacity of the internal state of the model.

Our application setting involves customer support agents consulting a proprietary knowledge base to resolve customer queries using specialized internal tools. The knowledge base contains documents vetted across multiple dimensions, including risk and legal compliance, making factual errors in the content highly unlikely. However, strict permissions govern which documents an agent can access. Figure 5(a) shows the complexity of document formats and fine-grained entitlements that impact retrieval and downstream generation.

We model confidence estimation as a classification problem over sequences of activations. Specifically, we employ a lightweight recurrent neural network (LSTM) that consumes FFN activations as auto-regressive signals. The classification logit from the LSTM head serves as the confidence score (see Figure 4). To enhance robustness against noisy supervision, we introduce a Huber loss regularizer L_{Huber} alongside the cross-entropy loss L_{CE}. The Huber loss's ability to behave quadratically for small errors and linearly for large errors makes it well-suited for smoothing gradients and mitigating the influence of outliers (Patra et al., 2023). Results in Table 1 demonstrate that our approach outperforms strong baselines, and the inclusion of L_{Huber} further improves accuracy over using L_{CE} alone.

In real-world deployment, retrieval-augmented generation (RAG) pipelines must meet strict latency requirements, as the LLM prompt length is constrained by model context limits and thousands of queries are processed daily. Tables 4 and 5 summarize our performance–latency trade-offs. Reducing the number of Llama 3.1 8B layers from 32 to 16 while keeping context size fixed preserves accuracy while reducing latency by approximately 42.5%. When the context size is reduced, alethic errors increase due to incomplete retrieval, raising the model's masking rate (i.e., instances where no answer is returned due to low confidence). Nevertheless, the 16-layer configuration achieves comparable performance to the 32-layer setup at lower computational cost. We observe a slight improvement in response latency when hosting the model using vLLM inference compared to Hugging Face's inference API, likely due to vLLM's optimized memory management and continuous batching capabilities.

Overall, our approach leveraging FFN activations as auto-regressive signals, modeling them with an LSTM, and regularizing with L_{Huber} proves effective in long-form RAG settings. This method improves the trustworthiness of LLM-generated responses and holds strong potential for safe deployment in sensitive, domain-specific applications.

6 Limitations

Our work pushes the boundary of confidence estimation in retrieval-augmented generation (RAG) for sensitive domains, but several practical considerations remain. Ideally, a RAG system should generate both the response and its confidence score

in a single pass. In our current implementation, the confidence score requires a second run of the system, which introduces additional computational and latency overhead.

While this design choice enables deeper access to model internals, it also necessitates operating in a white-box setting, as the confidence model relies on activation signals from the LLM to assess correctness. Furthermore, the method is customized to the specific architecture of the target model, meaning that adaptation to other LLMs may require reconfiguration and retraining. These limitations also present opportunities for future research: integrating confidence estimation directly into the generation process, reducing computational cost, and developing architecture-agnostic approaches that preserve the performance benefits of activation-based probing methods.

A limitation of this study is that the dataset cannot be made publicly available. The data contains sensitive and proprietary information pertaining to internal financial tools and knowledge resources used by service agents within a financial institution. This restriction is mandated by internal data governance policies to protect confidential and regulated financial information.

References

AI@Meta. 2024. Llama 3 model card.

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.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. *arXiv preprint arXiv:2402.11756*.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. HHEM-2.1-Open.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. 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 predictive uncertainty quantification of free-form large language models. *arXiv preprint arXiv:2307.01379*.

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. 2024. Enhancing confidence expression in large language models through learning from past experience. *arXiv preprint arXiv:2404.10315*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *arXiv preprint arXiv:2404.15993*.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Rishabh Patra, Ramya Hebbalaguppe, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. 2023. Calibrating deep neural networks using explicit regularisation and dynamic data pruning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1541–1549.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. 2023. Llamas know what gpts don't show: Surrogate models for confidence estimation. *arXiv preprint arXiv:2311.08877*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.