

# Résumé de l'article : *Can LLMs Express Their Uncertainty?*

Synthèse basée sur Xiong *et al.*, ICLR 2024

## 1 Problématique

Les grands modèles de langage (LLMs) tels que GPT-3.5, GPT-4 ou LLaMA-2 génèrent des réponses textuelles cohérentes mais sans indiquer leur degré d'incertitude. Cette incapacité à estimer la confiance compromet leur fiabilité dans des contextes critiques (raisonnement, éthique, décision automatisée).

Les approches existantes pour la calibration de confiance s'appuient sur un accès interne au modèle (*white-box*), utilisant par exemple les probabilités de tokens ou le fine-tuning, ce qui n'est pas possible pour les modèles commerciaux fermés. Ainsi, la question centrale posée par les auteurs est :

**Comment éliciter et quantifier la confiance d'un LLM dans un cadre purement *black-box* ?**

## 2 Méthodologie

Cette section détaille le cadre proposé par Xiong *et al.* (ICLR 2024) pour l'élicitation de la confiance en boîte noire (*black-box confidence elicitation*). Leur approche repose sur trois composantes principales : (1) le **prompting**, (2) le **sampling**, et (3) l'**agrégation**. Chaque combinaison de ces modules forme un algorithme d'estimation de confiance applicable à tout LLM fermé.

## 2.1 Stratégie de Prompting : verbaliser la confiance

L’objectif du prompting est d’inciter le modèle à exprimer explicitement son niveau de confiance sous forme textuelle (0–100%). Les auteurs évaluent cinq variantes, inspirées du raisonnement humain :

- (a) **Vanilla Prompt** — Demande directe : *“Lis la question, donne ta réponse et ta confiance en cette réponse.”*
- (b) **Chain-of-Thought (CoT)** — Encourage un raisonnement étape par étape avant la réponse finale : *“Explique ta réflexion pas à pas, puis donne ta réponse et ta confiance.”*
- (c) **Self-Probing** — Le modèle s’auto-évalue dans une seconde session : *“Voici ta réponse précédente. Quelle est la probabilité qu’elle soit correcte ?”*
- (d) **Multi-Step** — Le modèle segmente la question en sous-étapes  $S_i$ , attribue une confiance  $C_i$  à chaque étape, puis agrège :

$$C_{\text{global}} = \prod_{i=1}^n C_i$$

Cela réduit la surconfiance en forçant une introspection intermédiaire.

- (e) **Top-K Prompt** — Le modèle doit fournir ses  $K$  meilleures hypothèses, chacune associée à une probabilité.

Toutes les variantes partagent une note explicative ajoutée à la fin du prompt : *“Note : la confiance indique la probabilité que votre réponse soit correcte.”* Cette standardisation vise à uniformiser la compréhension du terme “confiance” entre modèles.

## 2.2 Stratégie de Sampling : exploiter la variabilité des générations

Le **sampling** consiste à générer plusieurs réponses à partir du même modèle pour estimer la stabilité de ses sorties. L’idée : si les réponses varient beaucoup, le modèle est incertain.

Trois approches principales sont explorées :

- (a) **Self-Random Sampling** — répéter le même prompt  $M$  fois avec une température non nulle. Cela exploite la stochasticité naturelle du

modèle pour obtenir des variations de réponses.

- **(b) Prompt Paraphrasing** — reformuler la question (paraphrases automatiques) afin de voir si les réponses restent cohérentes.
- **(c) Misleading Sampling** — injecter volontairement un indice erroné (e.g. “Je pense que la réponse est 12”) et observer si le modèle s’y laisse influencer. Les modèles très confiants restent stables ; les modèles incertains changent facilement d’avis.

## 2.3 Méthodes d’Agrégation : interpréter et combiner les signaux de confiance

Les **méthodes d’agrégation** constituent la dernière étape du pipeline. Elles ont pour objectif de fusionner les signaux de confiance issus de plusieurs générations (*sampling*) ou de plusieurs hypothèses (*Top-K prompting*).

### 2.3.1 Agrégation par Cohérence (*Consistency*)

La méthode la plus simple repose sur l’hypothèse que la stabilité du modèle face à des entrées similaires est corrélée à sa confiance.

$$C_{\text{consistency}} = \frac{1}{M} \sum_{i=1}^M \mathbb{I}\{Y_i = Y^*\}$$

où  $M$  est le nombre de générations et  $\mathbb{I}\{Y_i = Y^*\}$  vaut 1 si la  $i$ -ème réponse correspond à la réponse majoritaire  $Y^*$ , et 0 sinon.

### 2.3.2 Agrégation par Pondération de Confiance (*Average Confidence, Avg-Conf*)

La deuxième approche introduit la confiance verbalisée  $P_i$  (entre 0 et 1) dans le calcul. Plutôt que de compter toutes les générations de manière égale, elle accorde plus de poids à celles où le modèle s’est déclaré plus sûr :

$$C_{\text{avg-conf}} = \frac{\sum_{i=1}^M \mathbb{I}\{Y_i = Y^*\} \times P_i}{\sum_{i=1}^M P_i}$$

### 2.3.3 Agrégation par Classement de Paires (*Pair-Rank*)

Cette approche exploite les classements produits par le *Top-K prompting*, où le modèle génère plusieurs hypothèses ordonnées selon leur probabilité interne. Chaque génération  $i$  fournit une séquence ordonnée :

$$S^{(i)} = (S_1^{(i)}, S_2^{(i)}, \dots, S_K^{(i)}),$$

où  $S_1^{(i)}$  désigne la réponse préférée du modèle.

L'idée centrale est de ne pas se fier aux pourcentages de confiance déclarés, souvent biaisés, mais plutôt à **l'ordre de préférence** entre les réponses. On définit une relation stricte de préférence :

$$S_u \succ S_v \iff P(S_u) > P(S_v),$$

et on suppose que la probabilité d'observer cette préférence suit un modèle de type Bradley-Terry :

$$\Pr(S_u \succ S_v) = \frac{P(S_u)}{P(S_u) + P(S_v)}.$$

L'objectif est alors d'estimer la distribution  $P(S)$  sur l'ensemble des hypothèses  $\mathcal{S}$  en maximisant la vraisemblance des préférences observées à travers les différentes générations :

$$\min_P - \sum_{i=1}^N \sum_{S_u, S_v \in \mathcal{S}} \mathbb{I}\{S_u^{(i)} \succ S_v^{(i)}\} \log \frac{P(S_u)}{P(S_u) + P(S_v)} \quad \text{s.c.} \quad \sum_{S \in \mathcal{S}} P(S) = 1.$$

**Intuition.** Même si les probabilités déclarées par le modèle sont souvent mal calibrées, les **ordres de préférence relatifs** sont en général plus fiables. Ainsi, si une réponse  $S_u$  est presque toujours classée avant une autre  $S_v$ , on peut inférer que  $P(S_u) > P(S_v)$  avec un haut degré de confiance. À l'inverse, si les classements sont instables entre plusieurs générations, leurs probabilités associées  $P(S_u)$  et  $P(S_v)$  se rapprochent, traduisant une incertitude accrue.

Cette méthode produit donc une **distribution continue et calibrée** de probabilités à partir des préférences ordinales implicites du modèle, exploitant la stabilité de ses choix plutôt que ses scores verbalisés.

## 2.4 Méthodes *white-box* basées sur les probabilités de tokens

Bien que l’objectif principal de l’article soit de développer un cadre d’élicitation de confiance en mode *black-box*, les auteurs évaluent également trois méthodes *white-box* reposant sur l’accès aux **logits** du modèle, c’est-à-dire aux probabilités internes des tokens de sortie. Ces approches ne sont donc possibles que pour les modèles ouverts ou lorsqu’on dispose d’un accès direct aux distributions générées par le modèle.

On note  $y = (t_1, \dots, t_L)$  la séquence de tokens produite, et  $p(t_i \mid t_{<i})$  la probabilité (issue des logits) du token  $t_i$ .

**1. Sequence Probability (seq-prob).** Cette méthode utilise la probabilité totale de la séquence, définie comme le produit des probabilités des tokens :

$$P_{\text{seq}}(y) = \prod_{i=1}^L p(t_i \mid t_{<i}).$$

Ou équivalamment, en log-probabilité :

$$\log P_{\text{seq}}(y) = \sum_{i=1}^L \log p(t_i \mid t_{<i}).$$

Elle reflète la vraisemblance brute de la séquence générée, mais favorise fortement les réponses courtes.

**2. Length-Normalized Sequence Probability (len-norm-prob).** Pour corriger le biais lié à la longueur, les auteurs normalisent la log-probabilité par le nombre de tokens :

$$P_{\text{len-norm}}(y) = \exp\left(\frac{1}{L} \sum_{i=1}^L \log p(t_i \mid t_{<i})\right).$$

Cette métrique reflète la « probabilité moyenne par token », réduisant l’avantage mécanique des réponses courtes.

**3. Key Token Probability (token-prob).** Cette méthode se concentre uniquement sur les **tokens pertinents pour la réponse** (ex. le nombre final dans une question mathématique), afin d’éviter que la log-probabilité soit dominée par les parties explicatives :

$$P_{\text{key}} = \prod_{t_i \in \mathcal{K}} p(t_i \mid t_{<i}),$$

où  $\mathcal{K}$  est l’ensemble des tokens clés (ex. “35” dans la sortie : « Explanation : ... Answer : 35 »).

Cette approche cherche à mesurer l’incertitude *spécifiquement sur la réponse finale*, plutôt que sur tout le texte généré.

## 2.5 Métriques d’Évaluation de la Confiance

Les auteurs utilisent plusieurs métriques complémentaires pour évaluer la qualité des scores de confiance.

### 2.5.1 Expected Calibration Error (ECE)

L’ECE évalue la **calibration** du modèle, c’est-à-dire la cohérence entre la confiance déclarée et la probabilité réelle de réussite. Un modèle bien calibré est correct environ 80 % du temps lorsqu’il déclare une confiance de 80 %. Inversement, un écart entre confiance et précision réelle traduit une sur- ou sous-confiance.

Formellement, soit un ensemble de  $n$  questions  $\{x_i\}_{i=1}^n$ , pour lesquelles le modèle fournit une prédiction assortie d’un score de confiance  $c_i \in [0, 1]$ . On regroupe ces exemples en  $B$  intervalles de confiance (*bins*)  $S_b$ , tels que chaque bin contienne les échantillons dont la confiance  $c_i$  appartient à une plage spécifique (par exemple  $[0.0, 0.1)$ ,  $[0.1, 0.2)$ , ...,  $[0.9, 1.0]$ ).

Pour chaque bin  $S_b$ , on calcule :

$$\text{acc}(S_b) = \frac{1}{|S_b|} \sum_{i \in S_b} \mathbb{I}\{y_i = \hat{y}_i\} \quad \text{et} \quad \text{conf}(S_b) = \frac{1}{|S_b|} \sum_{i \in S_b} c_i,$$

où  $\text{acc}(S_b)$  est la proportion de prédictions correctes et  $\text{conf}(S_b)$  la confiance

moyenne déclarée.

L'ECE est alors définie comme la moyenne pondérée des écarts absolus entre confiance et exactitude :

$$\text{ECE} = \sum_{b=1}^B \frac{|S_b|}{n} |\text{acc}(S_b) - \text{conf}(S_b)|.$$

Un ECE faible indique une bonne calibration : la confiance exprimée correspond bien à la probabilité empirique de réussite. Un ECE élevé signale une surconfiance (le modèle se surestime) ou une sous-confiance (le modèle se sous-estime). Dans la pratique, les travaux cités par les auteurs (guo2017calibration, naeini2015bayesian) utilisent  $B = 10$  intervalles également répartis.

### 2.5.2 Area Under ROC Curve (AUROC)

L'AUROC (Area Under the Receiver Operating Characteristic Curve) évalue la capacité du modèle à distinguer les bonnes réponses des mauvaises selon le score de confiance. La courbe ROC trace le *taux de vrais positifs* (TPR) en fonction du *taux de faux positifs* (FPR) pour différents seuils de décision :

$$\text{TPR} = \frac{\text{VP}}{\text{VP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{VN}}.$$

Un AUROC de 0.5 indique une séparation aléatoire, tandis qu'un score proche de 1.0 traduit une excellente capacité à attribuer de faibles confiances aux erreurs et de fortes confiances aux bonnes réponses.

### 2.5.3 AUPRC-Positive et AUPRC-Negative

L'AUPRC complète l'AUROC en cas de déséquilibre de données :

- **AUPRC-Positive (PR-P)** : capacité à identifier correctement les réponses justes.
- **AUPRC-Negative (PR-N)** : aptitude à reconnaître les réponses erronées.

### 3 Expérimentation et Résultats

Les auteurs évaluent systématiquement les méthodes d’estimation de confiance dans deux configurations distinctes : (1) un cadre *black-box*, au cœur de leur contribution, et (2) un cadre *white-box*, utilisé uniquement comme point de comparaison à titre de référence.

Les expériences couvrent cinq familles de tâches (raisonnement commun, arithmétique, raisonnement symbolique, droit, éthique) et cinq grands modèles (GPT-3, GPT-3.5, GPT-4, Vicuna-13B, LLaMA-2-70B), soit plus de quarante combinaisons de prompts, stratégies de sampling et méthodes d’agrégation.

#### 3.1 Performance des méthodes *black-box*

Les combinaisons de prompting, sampling et agrégation définies dans le cadre *black-box* sont évaluées selon quatre métriques (ECE, AUROC, AUPRC-P, AUPRC-N). Les résultats montrent plusieurs tendances fortes :

- **Surconfiance systématique.** Les LLMs verbalisent des confiances très élevées (souvent 80–100%), indépendamment de l’exactitude réelle.
- **Effet d’échelle.** GPT-4 surpasse les modèles plus petits, tant en calibration ( $\text{ECE} \approx 0.18$ ) qu’en détection d’erreurs ( $\text{AUROC} \approx 0.63$ ).
- **Apport du sampling.** L’utilisation de plusieurs générations ( $M=5$ ) améliore drastiquement la capacité à distinguer les réponses correctes des incorrectes, notamment en raisonnement arithmétique ( $\text{AUROC} \approx 0.92$  sur GSM8K).
- **Rôle des agrégateurs.** L’agrégation Pair-Rank optimise la calibration ( $\text{ECE} \approx 0.028$ ), tandis que Avg-Conf est plus efficace pour la détection d’échecs.

Ces éléments montrent que les signaux obtenus en boîte noire — verbalisation, stabilité inter-génération, classements Top-K — peuvent partiellement compenser l’absence d’accès aux log-probabilités internes.

### 3.2 Comparaison avec les méthodes *white-box*

Pour situer les performances en contexte, les auteurs comparent les approches *black-box* aux trois méthodes *white-box* fondées sur les probabilités de tokens (logits) décrites dans la section méthodologie.

Les résultats expérimentaux (Tables 5–6 du papier) montrent que :

- Les méthodes *white-box* performant légèrement mieux, en particulier **len-norm-prob** et **token-prob**.
- L'écart avec les méthodes *black-box* reste **modeste** : par exemple, un AUROC passant typiquement d'environ 0.52 (*black-box*) à environ 0.60 (*white-box*).
- Même avec accès aux logits, les performances restent faibles (AUROC souvent entre 0.5 et 0.6), montrant que l'incertitude sémantique n'est pas correctement capturée.

Cette analyse comparative renforce l'idée que l'accès aux probabilités internes ne résout pas le problème fondamental : les logits reflètent surtout l'incertitude syntaxique locale (prochain token), et non l'incertitude sur la validité sémantique globale de la réponse.

## 4 Conclusion

Cette étude propose un cadre systématique pour l'élicitation de la confiance en mode *black-box*, une problématique devenue centrale avec la généralisation des modèles fermés (GPT-3.5, GPT-4). En décomposant l'estimation en trois modules — prompting, sampling et agrégation — les auteurs montrent qu'il est possible d'obtenir des signaux d'incertitude exploitables sans accès aux probabilités internes.

Les résultats révèlent cependant plusieurs limites structurelles :

- les LLMs expriment une **surconfiance marquée**, reproduisant des schémas linguistiques humains plus que de véritables estimations probabilistes ;
- les stratégies de sampling améliorent la détection d'erreurs, mais la **calibration reste imparfaite** même pour GPT-4 ;
- les méthodes *white-box*, bien qu'un peu meilleures, ne dépassent pas

- non plus le seuil de performance attendu, ce qui montre que l'accès aux logits n'est pas suffisant pour capter l'incertitude sémantique ;
- aucune méthode n'atteint des performances fiables sur les tâches spécialisées (droit, éthique), où l'AUROC demeure proche du hasard.

Ainsi, l'incertitude dans les LLMs apparaît comme un problème encore largement **ouvert**. Les auteurs recommandent, pour la pratique, une combinaison robuste et peu coûteuse :

Top-K prompting + Self-Random sampling + Avg-Conf (ou Pair-Rank).

Cette approche offre un compromis raisonnable entre efficacité, coût d'inférence et fiabilité relative.

Plus largement, ces travaux ouvrent la voie à de nouvelles recherches visant à relier les incertitudes lexicales, syntaxiques et sémantiques, et à développer des modèles mieux capables d'évaluer eux-mêmes la fiabilité de leurs propres réponses — une compétence indispensable pour une utilisation sûre et responsable des LLMs.