Kernel Token Contradiction: a Fast and Principled Approach for LLM Claim Uncertainty Quantification

Jérémie Dentan* **Alexi Canesse*** **Mahammed El Sharkawy** **Sonia Vanier**

LIX (École polytechnique, IP Paris, CNRS)
jeremie.dentan@polytechnique.edu

Abstract

Claim-level Uncertainty Quantification (UQ) aims to mitigate the lack of reliability of Large Language Models (LLMs) by evaluating the factuality of each claim in their outputs. We introduce Kernel Token Contradiction (KTC), a lightweight approach to compute claim-level UQ under realistic white-box conditions. KTC represents the candidate tokens involved in LLM generation as a positive semi-definite kernel that integrates both the LLM's conditional distribution and a token contradiction score. We then use the Von Neumann entropy to quantify the uncertainty of this kernel. To estimate token contradiction, we develop a new approach based on frequency statistics from the Wikipedia corpus. Although CPU-only, our approach achieves over an 8.2× speedup compared to state-of-the-art GPU-accelerated methods based on cross-encoders, and over a 65× speedup compared to CPU-only methods with comparable performance. Our evaluation spans two benchmarks across four European languages and 16 different models. KTC not only matches the average performance of existing methods but also outperforms them in high-precision regimes. This combination of computational efficiency and accuracy makes real-time monitoring of LLM outputs practical in production.

1 Introduction

Despite their wide adoption, the reliability of Large Language Models (LLMs) remains an open challenge, as their outputs are prone to contain plausible yet non-factual content, known as hallucinations [16, 18, 27, 40]. Uncertainty Quantification (UQ) aims to provide LLM users with an estimate of the model's confidence, with the objective of mitigating the negative consequences of factuality hallucinations. In particular, claim-level UQ [7, 9, 32] provides a fine-grained view of the factuality of each claim in LLMs' outputs. This enables the user to be cautious about the uncertain parts without rejecting the whole answer, which is costly in terms of computation and user experience.

State-of-the-art claim-level UQ methods are known to have major limitations, highlighted by the MUCH benchmark [7]. First, their recall in high-precision regimes is insufficient. For instance, top-performing methods such as CCP [9] only achieve an average precision of 69% at recall $\leq 20\%$, which is largely insufficient to effectively mitigate hallucinations. Second, they rely on Natural Language Inference (NLI) models to estimate contradiction. These models not only require additional GPU resources, but they are also too slow for realistic monitoring of LLMs' outputs in production, representing up to over 100% of the LLM generation runtime. Finally, these methods rely on NLI models trained on English corpora, which significantly alters their performance in other languages [7].

To mitigate these limitations, we developed a principled approach to detect contradictions between the candidate tokens emerging at each Next Token Prediction (NTP) step. Manual exploration revealed that these contradictions specifically arise in factuality hallucinations. For instance, when completing

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

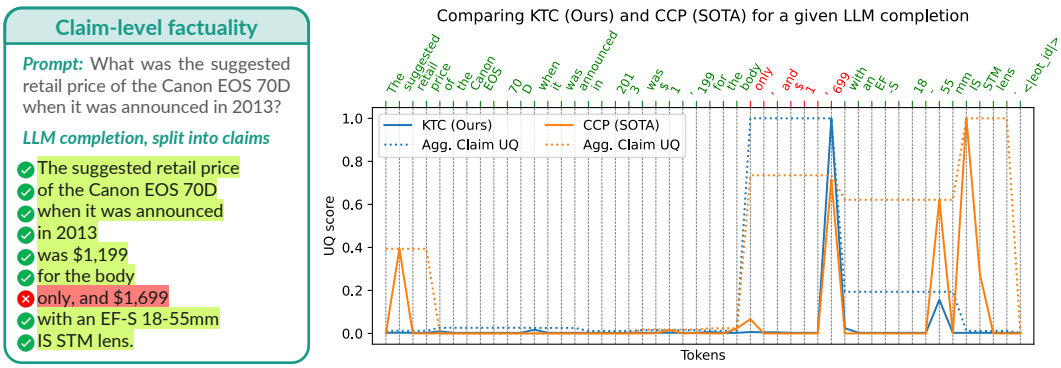


Figure 1: Example from the MUCH dataset [7]: only one of the nine claims in the LLM generation is incorrect. KTC correctly identifies the wrong price ($1,699), yielding a high uncertainty score for the incorrect claim, while CCP [9] produces several false positives and thus lower precision. Our method computes token-level scores that are aggregated at the claim level (see the dashed lines on the plot).

`"and $1,"` with NTP (see Figure 1), if the model computes a high probability for both tokens `"699"` and `"499"`, then it is uncertain about the price because it hesitates between two pieces of contradicting information. On the contrary, when completing `"$1,699 with"`, if the model hesitates between `"an"` and `"the"`, these two possibilities do not contradict each other and the model simply hesitates between different formulations. Based on this intuition, we developed a new heuristic based on neighbor frequency statistics in the Wikipedia corpus to detect contradicting tokens. This heuristic is orders of magnitude faster than the NLI models used in top-performing methods such as SAR [8] or CCP [9]. Moreover, its setup is much simpler than training an NLI model, requiring only the tokenization of many Wikipedia pages, which only takes a few minutes of CPU computation.

KTC consists of four steps, presented in Figure 2. It combines predictive uncertainty, derived from the entropy of the LLM output distribution, and semantic uncertainty, derived from contradictions between the meanings of candidate tokens. Steps [B] and [D] are inspired by Kernel Language Entropy (KLE) [25], a generation-level UQ method that represents generation outcomes as PSD kernels and computes their Von Neumann Entropy (VNE). Compared to existing methods, our pipeline introduces three major methodological innovations. First, we adapt the principle of KLE from the generation level to the token level. Second, we replace the NLI model used in CCP or SAR with the neighbor frequency heuristic introduced above (step [A] in Figure 2). Finally, we scale semantic uncertainty by predictive uncertainty (step [C] in Figure 2). This scaling is absent in KLE but plays a crucial role in the performance of our approach at the token level (see Section 3.3).

We evaluated KTC on two benchmarks (MUCH [7] and Mu-SHROOM [32]) spanning four languages (English, French, German, and Spanish) and 16 LLMs from various families. We compared KTC with comparable white-box claim-level UQ methods, including CCP [9] and SAR [8]. While CCP achieves the best overall performance, KTC ranks second with a small performance gap. Crucially, KTC is significantly more efficient, achieving at least an $8.2\times$ speedup and requiring no GPU acceleration. Moreover, KTC clearly outperforms all competing methods in the high-precision regime, achieving an average precision of 78% at recall $\leq 20\%$, compared to 69% for CCP. The computational efficiency and high accuracy of KTC make it practical for real-time monitoring of LLMs outputs.

Our contributions can be summarized as follows.

- We introduce a new heuristic based on token neighbor statistics to replace NLI models used in state-of-the-art claim-level UQ, with minimal computational and setup overhead;
- We propose a token-level score inspired by KLE [25], integrating the proposed heuristic and a scaling by the LLM predictive distribution that is critical for token-level UQ;
- We evaluate KTC on benchmarks covering four European languages and 16 LLMs, achieving comparable average performance to state-of-the-art methods while being substantially faster, and largely outperforming all methods in the high-precision regime;
- We release all source code and hyperparameters needed to reproduce our experiments.[1]

[1] See `https://github.com/orailix/kernel_token_contradiction` and Appendix E.

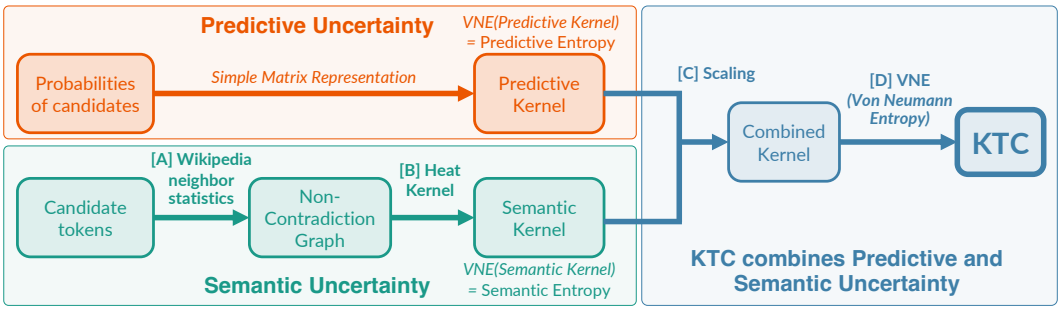


Figure 2: The KTC score is computed for each LLM-generated token. We construct a non-contradiction graph over candidate tokens (step [A]) and represent it as a PSD semantic kernel (step [B]). This kernel is then combined with the predictive kernel containing candidate probabilities (step [C]). Finally, we compute the Von Neumann entropy (VNE) of the combined kernel (step [D]).

2 Background

We briefly recall the uncertainty quantification tools and kernel methods principles underlying KTC.

Shannon's Entropy [28] Let $\delta \in \mathbb{N}^*$ and $p = \{p_i\}_{1 \leq i \leq \delta} \in [0,1]^\delta$ such that $\sum p_i = 1$ be a probability distribution over a finite set. Given a random variable $X \sim p$, Shannon's Entropy quantifies the degree of uncertainty of the system. It is defined as $H(X) = -\sum_{i=1}^{\delta} p_i \log p_i$, with the limit $x \log x \to 0$ as $x \to 0_+$. For instance, if $p_1 = 1$ and thus $p_{i>1} = 0$, the entropy is minimal and equal to zero; conversely, it is maximal and equal to $\log \delta$ when the p_i's are all equal to $1/\delta$.

Von Neumann Entropy (VNE) [33] Let $\rho \in \mathbb{R}^{\delta \times \delta}$ be a density matrix describing a quantum system, meaning that ρ is positive semi-definite and $\mathrm{Tr}(\rho) = 1$. Let $\lambda_1, \ldots, \lambda_\delta$ be the eigenvalues of ρ. VNE quantifies the quantum uncertainty of the system described by ρ. It is defined as

$$\mathrm{VNE} = -\mathrm{Tr}(\rho \log \rho) = -\sum_{i=1}^{\delta} \lambda_i \log \lambda_i. \tag{1}$$

Predictive Entropy Let $\mathcal{V} = \{t_1, \ldots, t_N\}$ be a vocabulary consisting of N distinct tokens, and let $f : \mathcal{V}^* \to [0,1]^N$ denote an LLM. It is trained using the Next Token Prediction (NTP) objective to approximate a target language modeling distribution P. For a given prefix sequence $x \in \mathcal{V}^*$, the i-th component of $f(x)$ estimates the probability that the *candidate token* t_i follows the prefix. Therefore, as training progresses, we aim to have the convergence $[f(x)]_i \longrightarrow P(t_i \mid x)$.

Let $\delta \in \mathbb{N}^*$ denote a maximum number of candidate tokens. For $i \in \{1, \ldots, \delta\}$, let $\Delta_i^\delta(x) \in \{1, \ldots, N\}$ denote the index of the i-th most likely candidate with respect to $f(x)$. Let $q_i := [f(x)]_{\Delta_i^\delta(x)}$ be the conditional probability of the i-th most likely candidate. We define the *predictive entropy* as Shannon's entropy of this discrete probability distribution. This quantity is also known as the *token entropy* in the literature [7, 14] and defined as

$$\text{Predictive Entropy} = -\sum_{i=1}^{\delta} \frac{q_i}{\sum_{1 \leq k \leq \delta} q_k} \log \frac{q_i}{\sum_{1 \leq k \leq \delta} q_k}. \tag{2}$$

Epistemic and Aleatoric Uncertainty A common distinction is made between *aleatoric uncertainty*, which captures the inherent randomness of LLM generations, and *epistemic uncertainty*, which arises from a lack of knowledge required to make accurate predictions [17]. In the context of LLMs, aleatoric uncertainty is desirable to ensure that the model remains creative, whereas epistemic uncertainty is associated with factuality hallucinations [36]. However, predictive entropy does not account for the *semantics* of tokens and therefore only captures the entropy of the *predictive distribution* over candidate tokens, without distinguishing between epistemic and aleatoric uncertainty. Drawing

inspiration from previous works [10, 20, 25], we propose a method that estimates the entropy of this distribution while accounting for semantic relations between tokens to isolate epistemic uncertainty.

3 Kernel Token Contradiction

In this section we introduce KTC, our new contradiction-based approach for computing claim-level uncertainty in LLM outputs. Our method consists of four stages, illustrated in Figure 2. For clarity, steps [B] and [D] are presented together in Section 3.2, while step [C] is presented in Section 3.3.

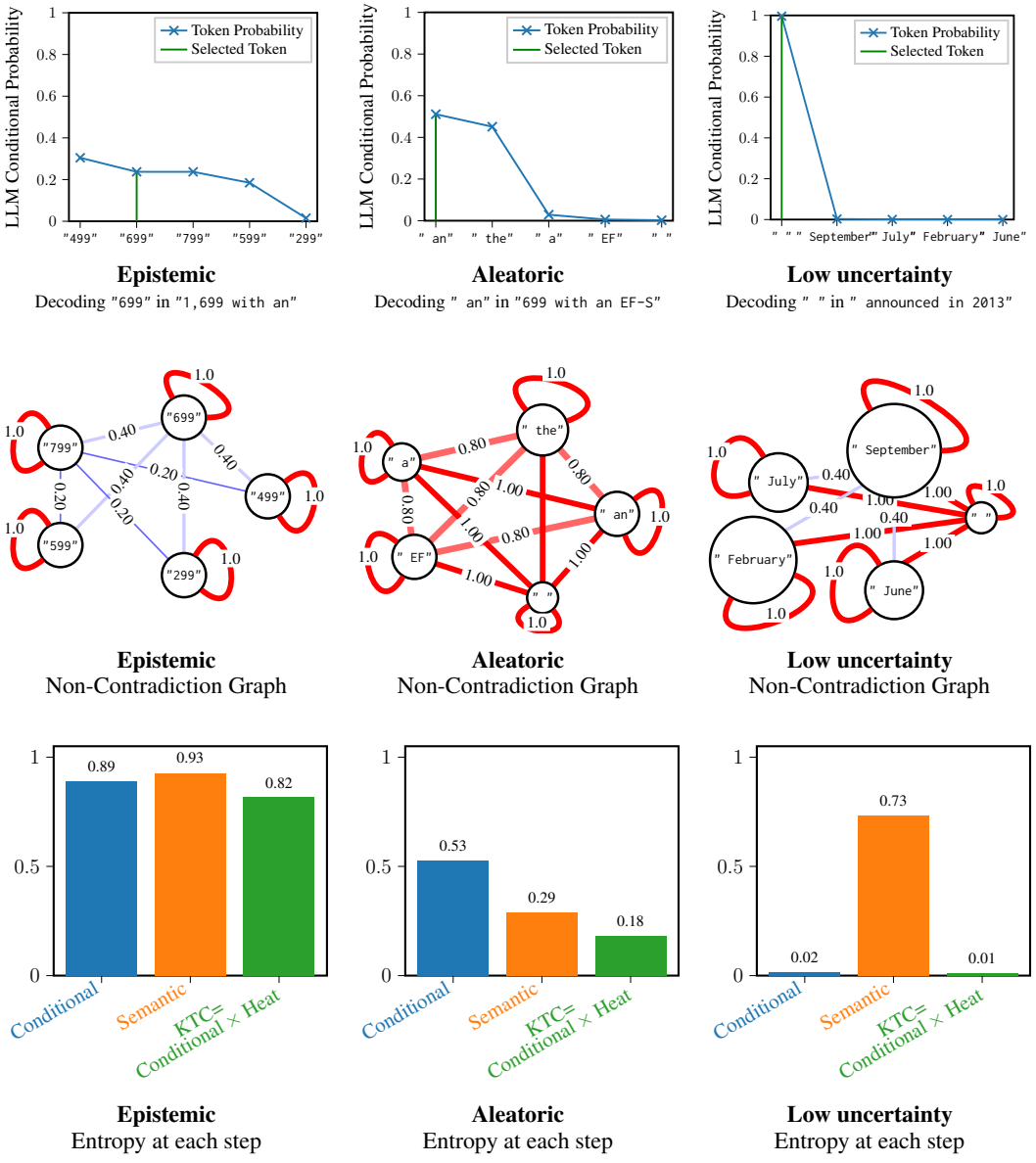


Figure 3: **Illustration of KTC computation for three tokens from the example in Figure 1. Upper:** candidate tokens and their likelihoods. **Middle:** the non-contradiction graph. **Lower:** the entropy at different stages of the computation. **Left:** epistemic uncertainty, where the model hesitates between contradicting prices, resulting in a sparse non-contradiction graph and a high KTC score. **Center:** aleatoric uncertainty, where the model hesitates between non-contradicting formulations, yielding high conditional entropy but low semantic entropy and thus a low KTC score. **Right:** low uncertainty, with low conditional entropy; despite high semantic entropy, scaling leads to a low KTC score.

3.1 [Step A] Building a Non-Contradiction Graph over Candidate Tokens

First, we build a non-contradiction graph over the candidate tokens $\left\{\Delta_i^\delta(x), i \in \{1, \dots, \delta\}\right\}$. For brevity, we simply refer to them as the Δ_i's. This graph is undirected and edge-weighted. Each node corresponds to an index $i \in \{1, \dots, \delta\}$, and the edge weights are scalars in $[0, 1]$ representing the non-contradiction between candidates Δ_i and Δ_j. Previous methods rely on an NLI model to assess entailment between candidate tokens [8, 9, 25]. However, this model induces a high computational cost, which can sometimes even surpass that of the LLM generation itself [7].

We introduce an alternative, extremely fast approach based on the most common neighboring tokens on Wikipedia. For a given LLM and a set of languages, we first tokenize a large part of the Wikipedia corpus. Then, for any candidate Δ_i and token $t \in \mathcal{V}$, we count how many times t is a neighbor of Δ_i, meaning that it appears right before or right after Δ_i in the corpus. Then, for $\nu \in \mathbb{N}^*$, we define $\mathcal{N}^\nu(\Delta_i) \subset \mathcal{V}$ as the ν most frequent neighbors of Δ_i. We use the number of common neighbors between candidates, denoted $|\mathcal{N}^\nu(\Delta_i) \cap \mathcal{N}^\nu(\Delta_j)|$, to define the non-contradiction score as

$$w_{i,j}(\nu) = 1 - \min\left[\frac{|\mathcal{N}^\nu(\Delta_i) \cap \mathcal{N}^\nu(\Delta_j)|}{\min\left(|\mathcal{N}^\nu(\Delta_i)|, |\mathcal{N}^\nu(\Delta_j)|\right)}, 1 - \mathbb{1}(\Delta_i \preceq \Delta_j \text{ or } \Delta_j \preceq \Delta_i)\right], \tag{3}$$

where $\Delta_i \preceq \Delta_j$ denotes that Δ_i is a prefix of Δ_j. Indeed, we consider that such tokens should have a non-contradiction score of 1 because the LLM can generate the missing suffix at the next NTP step.

Motivation and Validation We observe that two contradicting tokens such as `"699"` and `"499"` can replace each other in a text, so they will have similar neighboring sets, and a low non-contradiction score. Conversely, two tokens that are unrelated such as `" "` and `"September"`, or two tokens that entail each other such as `"an"` and `"the"`, will have different neighboring sets, and a non-contradiction score close to 1. The similarity of the neighboring sets indicates contradiction between tokens.

A potential limitation of this heuristic is that semantically related concepts, such as synonyms, hypernyms, or entities from the same category, may also have overlapping Wikipedia neighborhoods, resulting in an inappropriately low non-contradiction score. To test this hypothesis, we computed $w_{i,j}$ on 100 pairs of synonyms, of hypernyms, and of semantically close but non-hypernym word pairs. The results, reported in Appendix B, show that our heuristic assigns substantially higher non-contradiction scores to synonyms and hypernyms than to other semantically related concepts. It therefore captures non-contradiction rather than semantic similarity, as it was designed for.

3.2 [Steps B and D] Building the Semantic Kernel and Computing Von Neumann Entropy

Let $\mathcal{G}$ be the non-contradiction graph defined above, with nodes $\{1, \dots, \delta\}$ and adjacency matrix $\boldsymbol{A} = [w_{i,j}]_{1 \le i,j \le \delta}$. Let $\boldsymbol{D} \in \mathbb{R}^{\delta \times \delta}$ be the degree matrix of the graph. For any $\tau \in \mathbb{R}_+$, we compute the Laplacian $\boldsymbol{L}$, the *heat kernel* $\boldsymbol{\kappa}(\tau)$ of $\mathcal{G}$, and its normalized version $\boldsymbol{K}(\tau)$:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \quad ; \quad \boldsymbol{\kappa}(\tau) = \exp(-\tau \cdot \boldsymbol{L}) \quad ; \quad \boldsymbol{K}(\tau) = \boldsymbol{\kappa}(\tau) / \operatorname{Tr}(\boldsymbol{\kappa}(\tau)). \tag{4}$$

We refer to $\boldsymbol{K}$ as the *semantic kernel*, since it is built upon the semantics of candidate tokens via $\mathcal{G}$. Finally, we compute the Von Neumann entropy (VNE) of $\boldsymbol{K}(\tau)$ (see Section 2) to get the semantic entropy. Let $\lambda_1(\boldsymbol{K}), \dots, \lambda_\delta(\boldsymbol{K})$ be the eigenvalues of $\boldsymbol{K}(\tau)$. We define Semantic Entropy as:[2]

$$\text{Semantic Entropy} = \text{VNE}(\boldsymbol{K}(\tau)) = -\sum_{i=1}^{\delta} \lambda_i(\boldsymbol{K}) \cdot \log \lambda_i(\boldsymbol{K}). \tag{5}$$

Motivation The eigenvalues of the Laplacian are closely related to the connectivity of the graph. The Laplacian always has at least one eigenvalue equal to zero, associated with the eigenvector $(1, \dots, 1)$. The second-smallest eigenvalue of $\boldsymbol{L}$, also known as the Fiedler value, is equal to the *algebraic connectivity* of the graph, which reflects how well connected the graph is [12]. Here, the connectivity of the non-contradiction graph reflects semantic uncertainty: the more connected the graph is, the more the model hesitates between similar tokens, and therefore the lower the uncertainty.

[2] Note that this definition differs from that of Farquhar et al. [10], and only aims to distinguish it from Predictive Entropy.

Furthermore, let $\lambda_1 \leq \cdots \leq \lambda_\delta$ be the eigenvalues of $\boldsymbol{L}$. Then, the eigenvalues of $\boldsymbol{\kappa}(\tau)$ are given by $(e^{-\tau\lambda_1}, \ldots, e^{-\tau\lambda_\delta})$. Therefore, low semantic uncertainty leads to a high Fiedler value, meaning that all eigenvalues of $\boldsymbol{L}$ are large except the first one, which is zero. Consequently, $(e^{-\tau\lambda_1}, \ldots, e^{-\tau\lambda_\delta}) \simeq (1, 0, \ldots, 0)$, leading to a Von Neumann entropy close to zero. Conversely, high semantic uncertainty leads to a low Fiedler value, meaning that $e^{-\tau\lambda_1} \simeq e^{-\tau\lambda_2} \simeq 1$. Therefore, there are at least two similarly large eigenvalues of $\boldsymbol{K}$, leading to a high Von Neumann entropy.

Numerical Examples Consider a perfectly sparse non-contradiction graph with $\boldsymbol{A} = \boldsymbol{I}_\delta$. This corresponds to a very high semantic uncertainty, because all tokens contradict each other, with a non-contradiction score of zero. Therefore, we have $\boldsymbol{L} = 0$, $\boldsymbol{\kappa} = \boldsymbol{I}_\delta$, and $\lambda_1 = \cdots = \lambda_\delta = 1/\delta$. The Von Neumann entropy is then maximal, equal to $\log \delta$. Conversely, consider the scenario where $\mathcal{G}$ is a clique. This corresponds to very low semantic uncertainty, because tokens do not contradict each other at all. Then, the vectors $(1, -1, 0, \ldots, 0)$, $(1, 0, -1, 0, \ldots)$, $\ldots$, $(1, 0, \ldots, 0, -1)$ are eigenvectors of $\boldsymbol{L}$ associated with eigenvalue δ. Therefore, the eigenvalues of $\boldsymbol{\kappa}$ are $(1, e^{-\tau\delta}, \ldots, e^{-\tau\delta})$, and the Von Neumann entropy is close to zero for any non-negligible value of τ.

These two numerical examples illustrate that VNE is high in the case of high semantic uncertainty and low in the case of low semantic uncertainty, which is its intended behavior.

3.3 [Step C] Incorporating the Conditional Distribution

Let $\boldsymbol{\kappa}$ be the non-normalized *semantic* kernel defined above. Let $q_i \coloneqq [f(x)]_{\Delta_i}$ be the conditional likelihood of the i-th candidate token. The matrix $\boldsymbol{\Pi} \coloneqq \operatorname{diag}(q_i / \sum_k q_k)_{1 \leq i \leq \delta} \in \mathbb{R}^{\delta \times \delta}$ is a unit-trace PSD kernel representing the conditional likelihood of candidate tokens, without any semantic information. We compute a combined PSD kernel $\boldsymbol{\mu}$ that integrates both semantic and conditional kernels, and we use it to compute the final KTC value as the Von Neumann entropy:

$$\boldsymbol{\mu} = \boldsymbol{\Pi}^{1/2}\boldsymbol{\kappa}\boldsymbol{\Pi}^{1/2} \quad ; \quad \boldsymbol{M} = \boldsymbol{\mu} / \operatorname{Tr}(\boldsymbol{\mu}) \quad ; \quad \boxed{\text{KTC} = -\sum_{i=1}^{\delta} \lambda_i(\boldsymbol{M}) \cdot \log \lambda_i(\boldsymbol{M}).} \tag{6}$$

Motivation The heat kernel $\boldsymbol{\kappa}$ is a *diffusion operator*. Indeed, for any signal $u_0 \in \mathbb{R}^\delta$ defined on the graph, $u(\tau) = \boldsymbol{\kappa}(\tau)u_0$ is a solution of the diffusion equation $\partial u / \partial \tau + \boldsymbol{L}u = 0$. Therefore, with the eigendecomposition $\boldsymbol{\kappa} = \boldsymbol{V}^\top \boldsymbol{\Lambda} \boldsymbol{V}$, the eigenvectors contained in $\boldsymbol{V}$ are graph signals that are stable under the diffusion process. The matrix $\boldsymbol{\Pi}$ is an operator that scales a node signal according to the probability of each node. Thus, $\boldsymbol{\mu} = (\boldsymbol{V}\boldsymbol{\Pi}^{1/2})^\top \boldsymbol{\Lambda} (\boldsymbol{V}\boldsymbol{\Pi}^{1/2})$ scales these stable signals with respect to the conditional likelihood of each node, thereby incorporating $\boldsymbol{\Pi}$ into the semantic kernel.

Another interpretation is to observe that the eigenvalues of $\boldsymbol{\mu} = \boldsymbol{\Pi}^{1/2}\boldsymbol{\kappa}\boldsymbol{\Pi}^{1/2}$ are the same as those of $\boldsymbol{\kappa}\boldsymbol{\Pi}$. Yet, $\boldsymbol{\kappa}\boldsymbol{\Pi}$ is the composed operator of scaling by the predictive distribution and diffusion. This composition acts as a gating mechanism. For instance, if either $\boldsymbol{\Pi}$ or $\boldsymbol{\kappa}$ has rank 1, then the rank of $\boldsymbol{\kappa}\boldsymbol{\Pi}$ is also equal to 1 and the eigenvalues of $\boldsymbol{\kappa}\boldsymbol{\Pi}$ are all zero except one, leading to an entropy of zero. This means that either no conditional uncertainty ($\operatorname{rank}(\boldsymbol{\Pi}) = 1$, such as $q_1 = 1$ and $q_{i>1} = 0$, similar to column 3 in Figure 3) or no semantic uncertainty ($\operatorname{rank}(\boldsymbol{\kappa}) = 1$, such as $\boldsymbol{A} = \mathbf{1}$, the all-one matrix, similar to column 2 in Figure 3) leads to low KTC. The final entropy can be high only if both conditional entropy and semantic entropy are high (similar to column 1 in Figure 3).

4 Related Works

Factual, White-Box, and Instance-Specific UQ We focus exclusively on factuality hallucinations in LLM outputs, as opposed to *input-conflicting*, *context-conflicting*, or *faithfulness* hallucinations [16, 40]. Moreover, we focus on *white-box* approaches that require access to the model itself, with its logits and activations [4, 8, 9, 19, 29], as opposed to *black-box* approaches that only rely on the generated text [20, 21, 25]. Finally, our approach is instance-specific as it computes an uncertainty score for a given LLM generation, as opposed to population-specific approaches that assign a score to a group of generations answering the same prompt [20, 25, 29].

Comparison with Kernel-Based and Semantic Methods Our work builds upon kernel methods and semantic similarities for UQ in LLMs. Nikitin et al. [25] introduce Kernel Language Entropy (KLE), a generation-level method constructing PSD kernels from semantic similarities between outputs, computing their Von Neumann entropy. However, KLE operates at the *generation level* as do the works that built upon it, such as SGPU [15] or SNNE [24]. In contrast, KTC adapts this principle to the *token level*, enabling claim-specific UQ. Furthermore, KTC replaces the NLI models with a lightweight Wikipedia neighbor frequency heuristic, and integrates the predictive uncertainty.

White-Box Claim-Level UQ Methods Used as Baselines In Section 5, we compare KTC to other white-box methods adapted for claim-level UQ. As done in [7], we include CCP [9] and SAR [8], both of which rely on Natural Language Inference (NLI) scores. We also evaluate likelihood-based methods, including Maximum Likelihood [1], defined as the probability of the most likely token at each step, Token Likelihood [14], corresponding to the probability of the sampled token, and Token Entropy [23]. We did not include FOCUS [39], despite its strong performance on several benchmarks [26, 38], because it relies on attention scores, which are unavailable in modern implementations such as Flash-Attention [5, 6], making it impractical for production.

5 Empirical Results

Our evaluation shows that KTC achieves competitive performance with state-of-the-art methods while being significantly faster, making it practical for real-time monitoring of LLM outputs in production. We also show that KTC outperforms all baselines in the high-precision regime, which is the most relevant setting for real-world applications. Finally, we show that KTC generalizes well across languages, achieving competitive performance in all four languages tested.

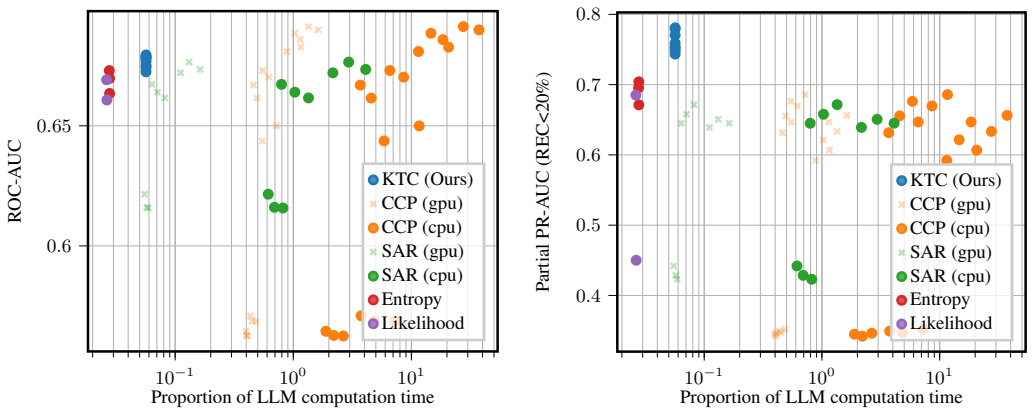
We evaluated our approach on two benchmarks (MUCH [7] and Mu-SHROOM [32]) spanning four languages (English, French, German, and Spanish) and 16 LLMs, including models from the Llama 3 [13], Mistral [22], Gemma 3 [30], Falcon [2], Pythia [3], Bloom [35], CroissantLLM [11], and Qwen 2 [37] families. We compared KTC with comparable white-box claim-level UQ methods, including CCP, SAR, Maximum Likelihood, Token Likelihood, and Token Entropy (see Section 4). All CPU experiments were run on an Apple M4 Pro (48 GB RAM), while GPU experiments were performed on an HPC node equipped with NVIDIA A100 GPUs and AMD EPYC 7543 CPUs.

***Main* and *Validation* Datasets** Our experiments focus on the high-precision/low-recall regime. Indeed, this regime is typically the most relevant for real-world deployment, where high precision is required to keep the number of false positives manageable [31, 34]. Consequently, to reduce noise due to annotation quality, we restrict our evaluations to the manually annotated subsets of MUCH and Mu-SHROOM in English, French, German, and Spanish. This forms our *main* evaluation dataset, with a total of 698 samples that are segmented into 9,359 claims using the claim segmenter provided in MUCH [7]. Moreover, to assess the generalization of KTC at a larger scale and on samples it was not tuned for, we also use the 4,673 samples of MUCH that were not annotated by humans as our *validation* set, comprising 19,884 claims.

Hyperparameters We evaluate all methods over predefined hyperparameter grids, summarized in Table 3 in Appendix D. Among the 30 configurations that we explored for KTC, we retained the best 8 according to ROC-AUC and PR-AUC. For CCP and SAR, we evaluated 18 and 9 parameter configurations that include three sizes of underlying NLI models (70M, 100M, and 350M) that we introduce to enable faster inference and assess whether this could improve their accuracy–efficiency trade-off relative to KTC. Hyperparameter generalization is discussed in Section 5.3.

5.1 Is Real-Time Monitoring of LLM Outputs Practical with KTC?

The speed of a UQ method is crucial for real-time monitoring applications. It should be quantified as a computational overhead relative to the LLM inference time [7]. If a method requires as much time or more than the LLM inference, its use becomes prohibitive in practice, as it would significantly increase the overall latency of the system. In contrast, a method with a negligible overhead can be easily integrated into existing pipelines without causing significant delays. To this end, we evaluate the performance of KTC and the baselines as a function of the computational overhead relative to the LLM inference time (see Figure 4). The overhead is computed as the ratio of the time taken by the



(a) ROC-AUC

(b) PR-AUC focusing on High Precision

Figure 4: **Accuracy-efficiency trade-off for white-box claim-level UQ methods.** Each point corresponds to a method evaluated under a different set of hyperparameters. The x-axis reports the UQ computation time as a fraction of the LLM generation cost. KTC achieves competitive accuracy at a fraction of the cost of NLI-based methods (CCP, SAR), while requiring no GPU.

UQ to the inference runtime, averaged over all samples, models, and languages. In this figure, we report the area under the curve (AUC) for the ROC curve, and for the precision-recall (PR) curve focusing on the high-precision regime (recall $\leq 20\%$). For NLI-based methods (CCP and SAR), we report the overhead for both CPU and GPU implementations, as these methods can be significantly accelerated by GPU computation. For KTC and other baselines, we report the overhead for the CPU implementation, as they do not require GPU acceleration. We include multiple points per method, corresponding to different hyperparameter sets (see details in Appendix D).

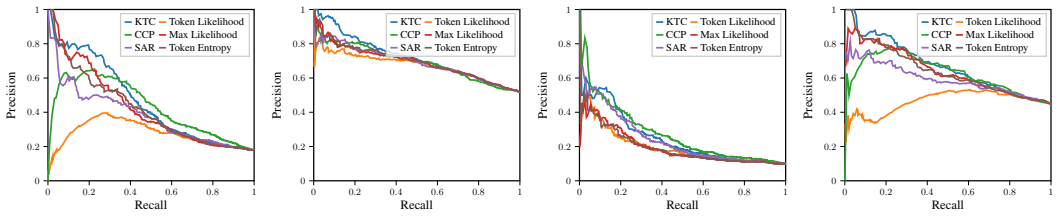
KTC achieves the best overall trade-off between performance and efficiency. In terms of ROC-AUC, the only baseline that surpasses KTC is CCP. However, KTC reaches its top performance with only a 5.6% overhead and runs entirely on CPU. In contrast, the CCP configurations that achieve comparable performance incur a 46% overhead on GPU ($8.2\times$ higher) and up to 369% on CPU ($65\times$ higher). For PR-AUC, KTC not only outperforms all baselines but does so by a clear margin in the high-precision regime. It achieves a PR-AUC of 0.78 at recall $\leq 20\%$, whereas CCP peaks at 0.69, despite its much higher cost. Moreover, the introduction of the 70M NLI variant for CCP does not improve the trade-off between performance and computational efficiency. This shows that the higher performance of CCP requires a high-performing NLI model, which incurs a high computational cost. Other methods further illustrate the favorable trade-off of KTC: SAR incurs a higher overhead ($\sim 10\%$ of the LLM generation cost on GPU) while delivering lower performance, and entropy- and likelihood-based baselines, although cheaper ($\sim 2.7\%$ overhead compared to LLM generation), remain behind in both ROC-AUC and PR-AUC. Overall, KTC is the only method that combines strong performance with consistently low computational cost.

5.2 Does KTC Generalize across Languages?

We compare the Precision-Recall curves of KTC and our baselines across four languages in Figure 5 (see ROC curves in Appendix C). To generate those figures, we evaluate all hyperparameter combinations and report results for the best hyperparameter set for each method and language. KTC consistently outperforms all baselines across languages in the low-recall, high-precision regime, which is typically the most relevant in production (see above). This trend holds in all languages and is particularly pronounced in English, French, and German, where KTC maintains a clear precision advantage over all baselines up to a recall of 0.4. In Spanish, the gap is narrower in the high-precision regime, and CCP surpasses KTC beyond 20% recall. We attribute this to a lower prevalence of hallucinations in Spanish, which makes the high-precision regime inherently more challenging.

Overall, KTC delivers competitive performance across all languages while consistently preserving its advantage in the high-precision regime. Notably, KTC relies on a Wikipedia-based non-contradiction

graph that is language-agnostic. In contrast, top-performing baselines such as SAR and CCP depend on NLI models trained in English, which can limit their ability to generalize across languages.



(a) English (b) French (c) Spanish (d) German

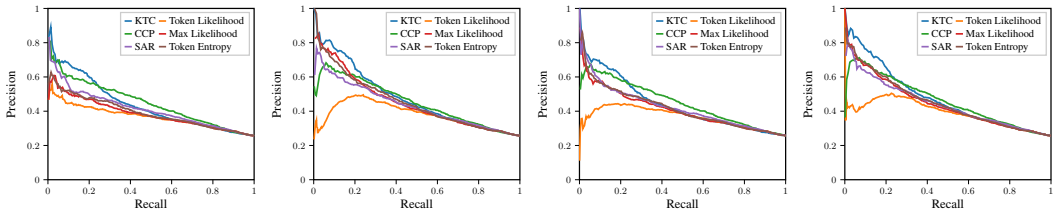
Figure 5: **Precision-Recall curves for claim-level hallucination detection** across four languages, using product as token-to-claim aggregation method. Results are averaged over 16 LLMs across two benchmarks. KTC notably outperforms competing methods in the high-precision regime (low recall), while remaining competitive overall.

5.3 Does KTC Generalize to the Validation Set?

We evaluated KTC and our baselines with various hyperparameters (see above). In Figure 4, we retain the 8 top-performing KTC configurations according to ROC-AUC and PR-AUC. Most use $\tau \in \{0.2, 0.3\}$ and $\nu \in \{3, 4, 5\}$, which we recommend as default values leading to similarly good performance. To assess hyperparameter generalization, we additionally evaluate these 8 configurations, without retuning, on our *validation* set introduced above. We compare them against all baseline configurations. Our results are reported in Figure 9 and Table 5. KTC maintains its favorable accuracy-efficiency trade-off on this unseen set, supporting the generalization of the selected hyperparameters at larger scale.

5.4 What is the Impact of Token-to-Claim Aggregation?

KTC produces an uncertainty score at each token position within a claim. To obtain a single claim-level score, we aggregate these per-position scores using one of four strategies: mean, maximum, geometric mean, or product over the token-level KTC values. Figure 6 compares KTC against all baselines under each strategy, averaged over all four languages and 16 LLMs. Further results are provided in Figure 8 in Appendix C. Across every aggregation strategy, KTC consistently dominates all baselines in the high-precision regime and achieves competitive ROC-AUC, confirming that the method's advantage is not contingent on a particular aggregation choice. We use the product aggregation as the default throughout the paper, as it is the one giving the best performance for all methods, aligning with the findings of Fadeeva et al. [9].



(a) Mean (b) Maximum (c) Geometric (d) Product

Figure 6: **Precision-recall curves under four token-to-claim aggregation strategies** (Mean, Maximum, Geometric mean, Product), averaged across four languages and 16 LLMs. Each method is evaluated over its full grid search space, and we report its best-performing configuration.

5.5 What is the Impact of the Semantic and Predictive Kernels?

We propose two approaches for merging the *predictive kernel* $\mathbf{\Pi}$ and the *semantic kernel* $\boldsymbol{\kappa}$:

$$\boldsymbol{\mu}^{\text{SCALE}} = \mathbf{\Pi}^{1/2}\boldsymbol{\kappa}\mathbf{\Pi}^{1/2} \qquad \text{and} \qquad \boldsymbol{\mu}_{\alpha}^{\text{COMB}} = \alpha\mathbf{\Pi} + (1-\alpha)\boldsymbol{\kappa}, \quad \alpha \in [0,1]. \tag{7}$$

In both cases, we divide the matrix by its trace to obtain $\boldsymbol{M}^{\text{SCALE}}$ and $\boldsymbol{M}_{\alpha}^{\text{COMB}}$ before computing the VNE, as in Equation (6). $\boldsymbol{M}^{\text{SCALE}}$ is simply denoted $\boldsymbol{M}$ in the rest of this paper because we use the SCALE method as the default for three reasons. First, it consistently outperforms COMB as evidenced in Table 1. Second, it does not rely on any hyperparameter. Finally, it can be interpreted intuitively as a scaling of the kernel $\boldsymbol{\kappa}$ by the conditional probabilities of the candidate tokens. In contrast, the convex combination involved in COMB is less interpretable.

The ablation of the α parameter for the COMB strategy in Table 1 provides insight into the respective roles of the semantic and predictive kernels. When $\alpha = 0$, the resulting kernel reduces to the purely semantic kernel and performs significantly worse. Increasing α improves performance, indicating that the conditional kernel plays a more dominant role. The fact that SCALE outperforms all COMB values shows that the predictive kernel alone is still insufficient, highlighting the benefit of the SCALE method for combining both components.

Table 1: **Precision-Recall AUC of aggregation methods** for varying neighborhood size ν. SCALE consistently outperforms COMB across all configurations.

ν	**SCALE**	**COMB**				
		$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
3	**0.471**	0.402	0.398	0.410	0.432	0.452
4	**0.476**	0.404	0.400	0.412	0.433	0.452
5	**0.472**	0.395	0.395	0.409	0.431	0.452
6	**0.467**	0.391	0.392	0.407	0.431	0.452
8	**0.457**	0.379	0.383	0.401	0.428	0.452

6 Limitations and Future Work

Our evaluation is limited to two manually annotated benchmarks (698 samples in total), which ensures high quality but remains relatively small-scale. We also restrict our experiments to English, French, German, and Spanish due to reliance on the MUCH segmenter [7]; extending to more languages would be a natural next step. Moreover, KTC does not dominate all baselines across all regimes, but instead provides a stronger performance–computation trade-off and consistently outperforms other methods in the high-precision setting, which is often the most relevant in practice.

Finally, KTC assumes access to token-level probabilities. Extending it to black-box settings with partial or approximate log-probabilities is an important direction for future work.

7 Conclusion

We introduced KTC, a token-level uncertainty quantification method that combines predictive uncertainty with a contradiction signal derived from Wikipedia neighbor statistics. By replacing the NLI models used in prior work with a lightweight corpus-based heuristic, KTC removes the dependency on GPU acceleration and English-trained inference models that limits existing approaches.

Empirically, KTC ranks second to CCP on overall ROC-AUC but achieves the best accuracy–cost trade-off by a wide margin, and clearly dominates all baselines in the high-precision regime that matters most for production hallucination mitigation. These results hold across four languages and 16 LLMs. While KTC requires white-box access to token probabilities and our evaluation covers a focused set of languages and benchmarks, its combination of CPU-only efficiency and high-precision accuracy makes it, to our knowledge, the first claim-level UQ method practical for real-time monitoring of LLM outputs under production latency constraints.

Acknowledgments and Disclosure of Funding

This work received direct financial support from Crédit Agricole SA through the research chair "Trustworthy and Responsible AI" with École polytechnique. This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014843R2 made by GENCI. We also thank Mohamed Dhouib and Mathis Le Bail for discussions on early versions of this paper.

Regarding competing interests, the authors declare ongoing research collaborations with Groupe SNCF and paid work by one author for Mistral AI within the 36 months preceding submission. These relationships did not influence the work presented in this paper.

Bibliography

[1] Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. "Rethinking Uncertainty Estimation in Natural Language Generation". In: *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The next Frontier in Reliable AI*. 2025. URL: https://openreview.net/forum?id=iBKWqXCSFA.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. *The Falcon Series of Open Language Models*. arXiv. Nov. 2023. DOI: 10.48550/arXiv.2311.16867.

[3] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. "Pythia: A Suite for Analyzing Large Language Models across Training and Scaling". In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org, 2023.

[4] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. "INSIDE: LLM's Internal States Retain the Power of Hallucination Detection". In: *International Conference on Learning Representations*. Vol. 2024. 2024, pp. 3056–3076. URL: https://proceedings.iclr.cc/paper_files/paper/2024/file/0d1986a61e30e5fa408c81216a616e20-Paper-Conference.pdf.

[5] Tri Dao. "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning". In: *International Conference on Learning Representations*. Vol. 2024. 2024, pp. 35549–35562. URL: https://proceedings.iclr.cc/paper_files/paper/2024/file/98ed250b203d1ac6b24bbcf263e3d4a7-Paper-Conference.pdf.

[6] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness". In: *Advances in Neural Information Processing Systems 35*. New Orleans, Louisiana, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022, pp. 16344–16359. ISBN: 978-1-7138-7108-8. DOI: 10.52202/068431-1189.

[7] Jérémie Dentan, Alexi Stanislas Canesse, Davide Buscaldi, Aymen Shabou, and Sonia Vanier. "MUCH: A Multilingual Claim Hallucination Benchmark". In: *The Fifteenth Language Resources and Evaluation Conference (LREC 2026)*. Palma, Mallorca, Spain, May 2026, pp. 2249–2267. DOI: 10.63317/4zrfhra4azck.

[8] Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. "Shifting Attention to Relevance: Towards the Predictive Uncertainty Quantification of Free-Form Large Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2024, pp. 5050–5063. DOI: 10.18653/v1/2024.acl-long.276.

[9] 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. "Fact-Checking the Output of Large Language Models via Token-Level Uncertainty Quantification". In: *Findings of the Association for Computational Linguistics ACL 2024*. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, 2024, pp. 9367–9385. DOI: 10.18653/v1/2024.findings-acl.558.

[10] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. "Detecting Hallucinations in Large Language Models Using Semantic Entropy". In: *Nature* 630.8017 (June 2024), pp. 625–630. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-024-07421-0.

[11] Manuel Faysse, Patrick Fernandes, Nuno M Guerreiro, António Loison, Duarte Miguel Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro Henrique Martins, Antoni Bigata Casademunt, François Yvon, Andre Martins, Gautier Viaud, CELINE HUDELOT, and Pierre Colombo. "CroissantLLM: A Truly Bilingual French-English Language Model". In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=uA19Xo1o31.

[12] Miroslav Fiedler. "Algebraic Connectivity of Graphs". In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305. ISSN: 0011-4642, 1572-9141. DOI: 10.21136/CMJ.1973.101168.

[13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, et al. *The Llama 3 Herd of Models*. arXiv. Nov. 2024. DOI: 10.48550/arXiv.2407.21783.

[14] Nuno M. Guerreiro, Elena Voita, and André Martins. "Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2023, pp. 1059–1075. DOI: 10.18653/v1/2023.eacl-main.75.

[15] Joseph Hoche, Andrei Bursuc, David Brellmann, Gilles Louppe, Pavel Izmailov, Angela Yao, and Gianni Franchi. *Improving Semantic Uncertainty Quantification in LVLMs with Semantic Gaussian Processes*. arXiv. Sept. 2026. DOI: 10.48550/arXiv.2512.14177.

[16] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: *ACM Transactions on Information Systems* 43.2 (Mar. 2025), pp. 1–55. ISSN: 1046-8188, 1558-2868. DOI: 10.1145/3703155.

[17] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods". In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-021-05946-3.

[18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (Dec. 2023), pp. 1–38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3571730.

[19] 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, et al. *Language Models (Mostly) Know What They Know*. arXiv. Nov. 2022. DOI: 10.48550/arXiv.2207.05221.

[20] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. "Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=VD-AYtP0dve.

[21] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. "Generating with Confidence: Uncertainty Quantification for Black-Box Large Language Models". In: *Transactions on Machine Learning Research* (2024). ISSN: 2835-8856. URL: https://openreview.net/forum?id=DWkJCSxKU5.

[22] Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, et al. *Ministral 3*. arXiv. Jan. 2026. DOI: 10.48550/arXiv.2601.08584.

[23] Andrey Malinin and Mark Gales. "Uncertainty Estimation in Autoregressive Structured Prediction". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=jN5y-zb5Q7m.

[24] Dang Nguyen, Ali Payani, and Baharan Mirzasoleiman. "Beyond Semantic Entropy: Boosting LLM Uncertainty Quantification with Pairwise Semantic Similarity". In: *Findings of the Association for Computational Linguistics: ACL 2025*. 2025, pp. 4530–4540. DOI: 10.18653/v1/2025.findings-acl.234.

[25] Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. "Kernel Language Entropy: Fine-Grained Uncertainty Quantification for LLMs from Semantic Similarities". In: *Advances in Neural Information Processing Systems*. Vol. 37. 2024, pp. 8901–8929. DOI: 10.52202/079017-0283.

[26] Elisei Rykov, Kseniia Petrushina, Maksim Savkin, Valerii Olisov, Artem Vazhentsev, Kseniia Titova, Alexander Panchenko, Vasily Konovalov, and Julia Belikova. "When Models Lie, We Learn: Multilingual Span-Level Hallucination Detection with PsiloQA". In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 11663–11682. DOI: 10.18653/v1/2025.findings-emnlp.626.

[27] Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. "A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024, pp. 11709–11724. DOI: 10.18653/v1/2024.findings-emnlp.685.

[28] Claude E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[29] Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. "LLM-Check: Investigating Detection of Hallucinations in Large Language Models". In: *Advances in Neural Information Processing Systems*. Vol. 37. 2024, pp. 34188–34216. DOI: 10.52202/079017-1077.

[30] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, et al. *Gemma 3 Technical Report*. arXiv. Mar. 2025. DOI: 10.48550/arXiv.2503.19786.

[31] Simon Valentin, Jinmiao Fu, Gianluca Detommaso, Shaoyuan Xu, Giovanni Zappella, and Bryan Wang. *Cost-Effective Hallucination Detection for LLMs*. arXiv. Aug. 2024. DOI: 10.48550/arXiv.2407.21424.

[32] Raul Vazquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sanchez-Vega, Alessandro Raganato, Jindřich Libovický, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona De Gibert, Jaione Bengoetxea, et al. "SemEval-2025 Task 3: Mu-SHROOM, the Multilingual Shared-Task on Hallucinations and Related Observable Overgeneration Mistakes". In: *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 2472–2497. ISBN: 979-8-89176-273-2. URL: https://aclanthology.org/2025.semeval-1.322/.

[33] John Von Neumann. *Mathematical Foundations of Quantum Mechanics*. Princeton university press, 1955.

[34] Song Wang, Xun Wang, Jie Mei, Yujia Xie, Si-Qing Chen, and Wayne Xiong. "Developing a Reliable, Fast, General-Purpose Hallucination Detection and Mitigation Service". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*. 2025, pp. 971–978. DOI: 10.18653/v1/2025.naacl-industry.72.

[35] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. arXiv. June 2023. DOI: 10.48550/arXiv.2211.05100.

[36] Yasin Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. "To Believe or Not to Believe Your LLM: Iterative Prompting for Estimating Epistemic Uncertainty". In: *Advances in Neural Information Processing Systems 37*. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, pp. 58077–58117. ISBN: 979-8-3313-1438-5. DOI: 10.52202/079017-1852.

[37] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, et al. *Qwen2 Technical Report*. arXiv. Sept. 2024. DOI: 10.48550/arXiv.2407.10671.

[38] Min-Hsuan Yeh, Max Kamachee, Seongheon Park, and Yixuan Li. "HalluEntity: Benchmarking and Understanding Entity-Level Hallucination Detection". In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=494k7e9R5D.

[39] Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. "Enhancing Uncertainty-Based Hallucination Detection with Stronger Focus". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 915–932. DOI: 10.18653/v1/2023.emnlp-main.58.

[40] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models". In: *Computational Linguistics* 51.4 (Dec. 2025), pp. 1373–1418. ISSN: 0891-2017, 1530-9312. DOI: 10.1162/COLI.a.16.

A Choice of Kernel

The choice of kernel is crucial to accurately capture semantic uncertainty. While we used a heat kernel in our method, we also experimented with other kernels, introduced below.

Let $\mathcal{G}$ be the non-contradiction graph defined above, with nodes $\{1, \dots, \delta\}$ and adjacency matrix $\boldsymbol{A} = [w_{i,j}]_{1 \leq i,j \leq \delta}$. Let $\boldsymbol{D} \in \mathbb{R}^{\delta \times \delta}$ be the degree matrix of the graph. Given that $\boldsymbol{A}$ is symmetric, we also consider its eigendecomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$, where $\boldsymbol{\Sigma}$ is a diagonal matrix containing the eigenvalues. We consider several approaches to build a PSD kernel upon $\boldsymbol{A}$:

$$\begin{cases} \boldsymbol{\kappa}^{\text{CLIP}} = \boldsymbol{U} \times \text{clip}(\boldsymbol{\Sigma}, \min = 0) \times \boldsymbol{U}^\top \\ \boldsymbol{\kappa}^{\text{AUGM}} = \boldsymbol{A} - \min(0, \min(\boldsymbol{\Sigma})) \times \boldsymbol{I}_\delta \\ \boldsymbol{\kappa}^{\text{SQR}} = \boldsymbol{A} \times \boldsymbol{A} \\ \boldsymbol{\kappa}^{\text{LAPL}} = \boldsymbol{D} - \boldsymbol{A} \\ \boldsymbol{\kappa}^{\text{HEAT}}(\tau) = \exp(-\tau \cdot [\boldsymbol{D} - \boldsymbol{A}]), \quad \tau \in]0, 1] \end{cases} \tag{8}$$

where $\text{clip}(\cdot)$ is an element-wise operations and $\min(\cdot)$ is a scalar denoting the minimum. For each of these kernels, we divide it by its trace, leading to the unit-trace PSD kernels $\boldsymbol{K}^{\text{CLIP}}$, $\boldsymbol{K}^{\text{AUGM}}$, $\boldsymbol{K}^{\text{SQR}}$, $\boldsymbol{K}^{\text{LAPL}}$, and $\boldsymbol{K}^{\text{HEAT}} = \boldsymbol{K}$. All of these kernels are built on the semantics of the candidate tokens via the non-contradiction graph, which is why we refer to these kernels as the *semantic* kernels.

In practice, preliminary experiments with KTC showed that the other kernels are less effective than the heat kernel at capturing semantic uncertainty, in line with the findings of Nikitin et al. [25]. Consequently, we exclude them from our experiments and focus solely on the heat kernel.

B Validation of the Wikipedia Heuristic

KTC requires a measure of whether two candidate tokens can be substituted without changing the factual content of the generated statement. We estimate this quantity from Wikipedia rather than using a neural semantic similarity model, which keeps the method lightweight and CPU-only. For each token, we construct a neighborhood from its occurrences in Wikipedia. The non-contradiction score between two candidate tokens is then computed from the overlap between their Wikipedia neighborhoods (see Equation (3)).

A potential limitation of this heuristic based on Wikipedia tokenization is that semantically related concepts, such as synonyms, hypernyms, or entities from the same category, may also have overlapping Wikipedia neighborhoods, resulting in an inappropriately low non-contradiction score. To test whether the heuristic captures non-contradiction rather than semantic similarity, we prompted a language model to construct three sets of 100 word pairs: (i) synonyms, e.g., *car–automobile*, (ii) hypernyms, e.g., *dog–animal*, and (iii) semantically close but non-hypernym concepts, e.g., *France–Germany*. The latter set covers ten categories outlined in Table 2. The full list of words is available in our repository at https://github.com/orailix/kernel_token_contradiction.

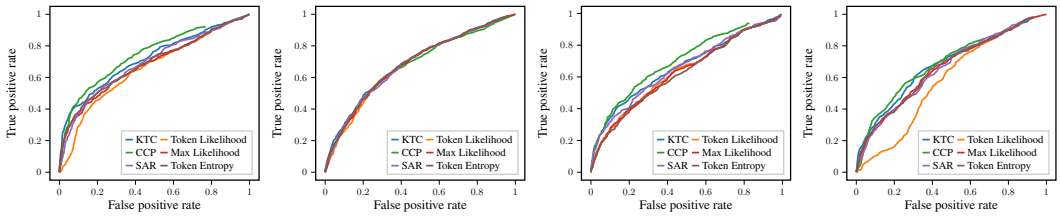
Our heuristic was designed to assign a low score to contradicting pairs of words where a substitution between the two words in a sentence would result in two contradicting statements. Therefore, we want our heuristic to assign low scores to semantically close, non-hypernym pairs, because words such as *France* and *Germany* are not substitutable. On the other hand, we want it to assign high scores to synonyms and hypernyms, because substituting a word like *car* with *automobile* or *vehicle* does not contradict the original sentence containing *car*. Our results in Table 2 confirm that this behavior is observed in practice, which supports the interpretation of our heuristic as a measure of substitutability rather than generic semantic similarity. Some categories, such as animals and space objects, nevertheless exhibit relatively high scores, reflecting the limitations of a corpus-based heuristic. However, KTC does not use these pairwise scores alone. Steps [B], [C], and [D], discussed in Section 3, aggregate the relationships between multiple candidate tokens, as well as their predictive distributions, resulting in a measure that is much more robust than individual token-pair scores (see empirical evaluations in Section 5).

Table 2: **Non-contradiction scores produced by the Wikipedia heuristic.** Synonyms and hypernyms receive substantially higher scores than semantically close but non-hypernym concepts, indicating that the heuristic does not simply measure generic semantic similarity. Values are mean $\pm$ standard deviation over the pairs of the category.

Pair type / category	**Non-contradiction**
Synonyms	0.860 ± 0.152
Hypernyms	0.837 ± 0.132
Semantically close, non-hypernym	0.574 ± 0.234
European countries	0.475 ± 0.146
Social media platforms	0.525 ± 0.146
Months	0.238 ± 0.153
Computer hardware	0.763 ± 0.205
Animals	0.825 ± 0.100
French cities	0.538 ± 0.202
Car models	0.475 ± 0.094
Programming languages	0.538 ± 0.217
Space objects	0.750 ± 0.209
Cocktails	0.613 ± 0.142

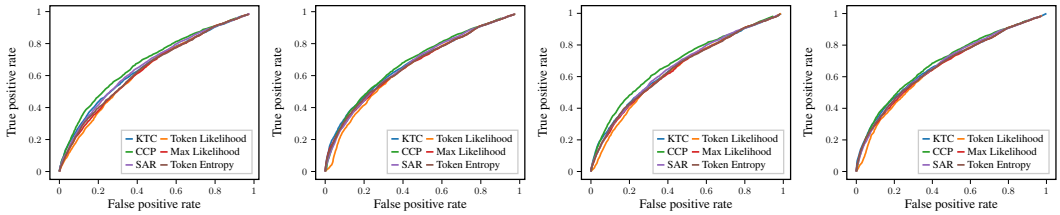
C ROC Curves

In addition to the Precision-Recall curves outlined in Section 5, we report the ROC curves for the top-performing hyperparameter configurations of each method. Figure 7 presents the top-performing results in each language (English, French, Spanish, and German), while Figure 8 presents the top-performing results depending on the token-to-claim aggregation strategy. KTC achieves the second-best performance, closely following CCP.



(a) English (b) French (c) Spanish (d) German

Figure 7: **ROC curves for claim-level hallucination detection** across four languages, using product for token-to-claim aggregation. Results are averaged over 16 LLMs across two benchmarks. Each method is evaluated over its full grid search space, and we report its best-performing configuration.



(a) Mean (b) Maximum (c) Geometric (d) Product

Figure 8: **ROC curves under four token-to-claim aggregation strategies** (Mean, Maximum, Geometric mean, Product), averaged over all four languages and 16 LLMs. Each method is evaluated over its full grid search space, and we report its best-performing configuration.

D Hyperparameter Tuning and Generalization

We evaluate all methods over predefined hyperparameter grids, summarized in Table 3. For KTC, this gives 30 configurations. We select the seven best configurations according to ROC-AUC and the seven best according to PR-AUC; after removing overlaps, this results in the eight configurations reported in the main paper.

For SAR and CCP, similar to what is done for the MUCH benchmark [7], we introduce a parameter σ that quantifies the number of preceding and succeeding tokens that are fed to the NLI model to compute the contradiction, entailment, or neutral relationship. We also introduce a Small variant using a 70M NLI model even though it was not considered in the original papers [8, 9], to enable faster inference and assess whether this could improve their accuracy–efficiency trade-off relative to KTC. Our experiments presented in Section 5 show that it does not yield a better trade-off than KTC. Overall, we evaluate 9 and 18 hyperparameter configurations for SAR and CCP, respectively.

- Small (70M parameters): `MoritzLaurer/DeBERTa-v3-xsmall-mnli-fever-anli-ling-binary`
- Medium (100M parameters): `microsoft/deberta-base-mnli`
- Large (350M parameters): `microsoft/deberta-large-mnli`

Table 3: Hyperparameter configurations evaluated for each method.

Method	**Hyperparameter**	**Values**
KTC	ν	$\{3, 4, 5, 6, 8\}$
	τ	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.7\}$
	δ	24 (fixed)
SAR	σ	$\{3, 5, 8\}$
	NLI model	Small, Medium, Large
CCP	σ	$\{3, 5, 8\}$
	δ	$\{10, 24\}$
	NLI model	Small, Medium, Large
Token Entropy	δ	$\{5, 10, 24\}$
Likelihood	Variant	Token, Max

D.1 Evaluation on the *Main* Dataset

Figure 4 reports the performance of the 8 best KTC configurations, as well as the performance of all configurations for the baselines. Time is reported relative to LLM generation time. For CCP and SAR, the same predictions are obtained on CPU and GPU; only the computation time differs. Detailed numerical values are provided in Table 4.

Table 4: **Performance of the evaluated hyperparameter configurations on our *main* evaluation set** (698 manually-annotated samples from MUCH [7] and Mu-SHROOM [32]). Time is relative to LLM generation time.

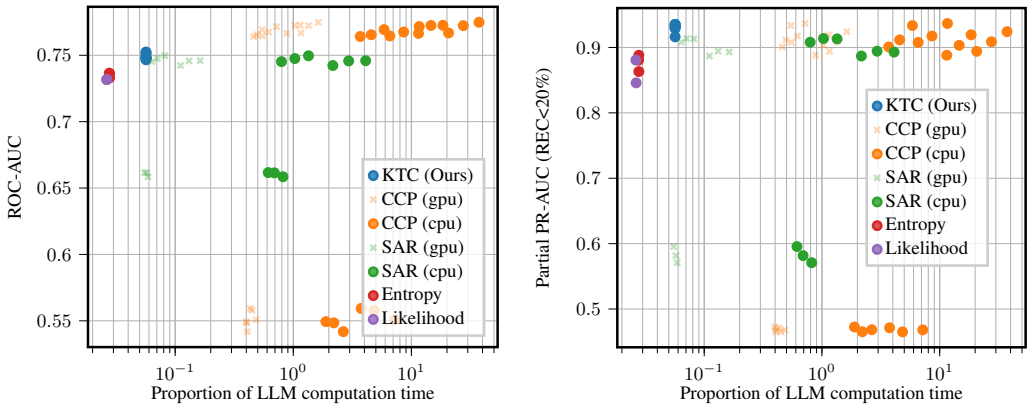
Method	Configuration	GPU time	CPU time	ROC-AUC	PR-AUC
KTC	$\nu = 3, \tau = 0.2$	–	0.0566	0.6795	0.7483
KTC	$\nu = 3, \tau = 0.3$	–	0.0566	0.6786	0.7590
KTC	$\nu = 4, \tau = 0.2$	–	0.0566	0.6787	0.7540
KTC	$\nu = 4, \tau = 0.3$	–	0.0566	0.6779	0.7808
KTC	$\nu = 4, \tau = 0.4$	–	0.0566	0.6725	0.7704
KTC	$\nu = 5, \tau = 0.2$	–	0.0566	0.6767	0.7515
KTC	$\nu = 5, \tau = 0.3$	–	0.0566	0.6742	0.7794
KTC	$\nu = 6, \tau = 0.2$	–	0.0566	0.6751	0.7434
CCP	$\delta = 10, \sigma = 3$, Large	0.8812	11.4870	0.6809	0.5923
CCP	$\delta = 10, \sigma = 3$, Medium	0.4611	3.6900	0.6670	0.6316
CCP	$\delta = 10, \sigma = 3$, Small	0.3981	1.8883	0.5644	0.3449
CCP	$\delta = 10, \sigma = 5$, Large	1.0260	14.6420	0.6885	0.6215
CCP	$\delta = 10, \sigma = 5$, Medium	0.4925	4.5712	0.6616	0.6555
CCP	$\delta = 10, \sigma = 5$, Small	0.4036	2.2030	0.5627	0.3420
CCP	$\delta = 10, \sigma = 8$, Large	1.1440	18.4894	0.6859	0.6469
CCP	$\delta = 10, \sigma = 8$, Medium	0.5445	5.8694	0.6437	0.6762
CCP	$\delta = 10, \sigma = 8$, Small	0.4076	2.6515	0.5625	0.3463
CCP	$\delta = 24, \sigma = 3$, Large	1.1589	20.6294	0.6828	0.6067
CCP	$\delta = 24, \sigma = 3$, Medium	0.5500	6.5696	0.6730	0.6468
CCP	$\delta = 24, \sigma = 3$, Small	0.4327	3.7530	0.5709	0.3493
CCP	$\delta = 24, \sigma = 5$, Large	1.3470	27.5059	0.6913	0.6334
CCP	$\delta = 24, \sigma = 5$, Medium	0.6184	8.5995	0.6703	0.6696
CCP	$\delta = 24, \sigma = 5$, Small	0.4477	4.8387	0.5686	0.3472
CCP	$\delta = 24, \sigma = 8$, Large	1.6263	37.3485	0.6900	0.6562
CCP	$\delta = 24, \sigma = 8$, Medium	0.7207	11.6444	0.6500	0.6857
CCP	$\delta = 24, \sigma = 8$, Small	0.4862	7.1833	0.5686	0.3516
SAR	$\sigma = 3$, Large	0.1109	2.1637	0.6721	0.6391
SAR	$\sigma = 3$, Medium	0.0637	0.7946	0.6673	0.6449
SAR	$\sigma = 3$, Small	0.0551	0.6137	0.6215	0.4420
SAR	$\sigma = 5$, Large	0.1314	2.9426	0.6765	0.6506
SAR	$\sigma = 5$, Medium	0.0708	1.0307	0.6640	0.6578
SAR	$\sigma = 5$, Small	0.0574	0.6924	0.6161	0.4286
SAR	$\sigma = 8$, Large	0.1629	4.0991	0.6735	0.6451
SAR	$\sigma = 8$, Medium	0.0818	1.3454	0.6616	0.6715
SAR	$\sigma = 8$, Small	0.0590	0.8183	0.6158	0.4230
Token Entropy	$\delta = 5$	–	0.0279	0.6635	0.6712
Token Entropy	$\delta = 10$	–	0.0279	0.6698	0.7040
Token Entropy	$\delta = 24$	–	0.0277	0.6730	0.6956
Likelihood	Token	–	0.0264	0.6608	0.4501
Likelihood	Max	–	0.0263	0.6692	0.6851

D.2 Generalization to the *Validation* Dataset

To assess whether the selected KTC hyperparameters generalize beyond the manually annotated data, we evaluate the eight selected configurations without retuning, on our *validation* set introduced in Section 5 and consisting of the 4,673 automatically annotated samples from MUCH [7], which are excluded from the experiments on the *main* set. We compare them with all configurations of SAR and CCP. Results are reported in Figure 9 and Table 5. The selected KTC configurations retain a competitive trade-off between performance and computational efficiency, with ROC-AUC ranging from 0.7466 to 0.7524 and PR-AUC from 0.9162 to 0.9354. Moreover, CCP and KTC are on par

in the high-precision regime (PR-AUC at recall $\leq 20\%$, with CCP at 0.9367 and KTC at 0.9354) on this *validation* set, whereas KTC outperforms CCP in this regime on the *main* evaluation set. However, the noise induced by the automated annotations of the *validation* set, compared to the human annotations of the *main* evaluation set, weakens these results, especially in the high-precision regime, where a few incorrect annotations can heavily impact the evaluation.

Overall, these results on the *validation* set show that KTC retains its competitive trade-off between performance and computational efficiency, as well as its strong results in the high-precision regime. This supports the generalization of the selected hyperparameters at larger scale, beyond the *main* evaluation set used for hyperparameter tuning.



(a) ROC-AUC

(b) PR-AUC focusing on High Precision

Figure 9: **Hyperparameter generalization.** Each point corresponds to a method evaluated under a different set of hyperparameters. The x-axis reports the UQ computation time as a fraction of the LLM generation cost. KTC configurations are selected on the manually annotated data and evaluated without retuning, on our validation set, showing that the accuracy-efficiency trade-off generalizes to unseen data. See detailed numbers in Table 5.

E Reproducible Science

Data and Code Availability The datasets used in this paper are publicly available. The MUCH dataset is available at `https://huggingface.co/datasets/orailix/MUCH`, and the Mu-SHROOM dataset is available at `https://huggingface.co/datasets/Helsinki-NLP/mu-shroom`. Alongside this paper, we release a code repository to reproduce our results.

Licenses In this paper, we use two datasets: MUCH and Mu-SHROOM. They are properly cited in the main content of the paper. The former is released under the Apache License 2.0 and the latter is released under the Creative Commons Attribution 4.0 license. Furthermore, NLI models are used in some of the baselines. We use the `microsoft/deberta-base-mnli`, `microsoft/deberta-large-mnli`, and `MoritzLaurer/DeBERTa-v3-xsmall-mnli-fever-anli-ling-binary` models, available through the MIT license.

Reproducibility All the results are completely reproducible. We release a code repository that includes the necessary information on library versions to reproduce our environment and run all the experiments and reproduce the results. Additional details on hyperparameter tuning are provided in Appendix D, and exact values are also included in the code repository.

`orailix/kernel_token_contradiction`

Table 5: **Performance of the evaluated hyperparameter configurations on our *validation* set** (4,673 automatically annotated samples from MUCH [7]). Time is relative to LLM generation time. The KTC configurations are evaluated without retuning.

Method	Configuration	GPU time	CPU time	ROC-AUC	PR-AUC
KTC	$\nu = 3, \tau = 0.2$	–	0.0566	0.7480	0.9354
KTC	$\nu = 3, \tau = 0.3$	–	0.0566	0.7503	0.9311
KTC	$\nu = 4, \tau = 0.2$	–	0.0566	0.7482	0.9349
KTC	$\nu = 4, \tau = 0.3$	–	0.0566	0.7524	0.9352
KTC	$\nu = 4, \tau = 0.4$	–	0.0566	0.7481	0.9162
KTC	$\nu = 5, \tau = 0.2$	–	0.0566	0.7476	0.9311
KTC	$\nu = 5, \tau = 0.3$	–	0.0566	0.7520	0.9336
KTC	$\nu = 6, \tau = 0.2$	–	0.0566	0.7466	0.9289
CCP	$\delta = 10, \sigma = 3$, Large	0.8812	11.4870	0.7665	0.8881
CCP	$\delta = 10, \sigma = 3$, Medium	0.4611	3.6900	0.7641	0.9008
CCP	$\delta = 10, \sigma = 3$, Small	0.3981	1.8883	0.5494	0.4725
CCP	$\delta = 10, \sigma = 5$, Large	1.0260	14.6420	0.7725	0.9033
CCP	$\delta = 10, \sigma = 5$, Medium	0.4925	4.5712	0.7655	0.9116
CCP	$\delta = 10, \sigma = 5$, Small	0.4036	2.2030	0.5484	0.4652
CCP	$\delta = 10, \sigma = 8$, Large	1.1440	18.4894	0.7726	0.9194
CCP	$\delta = 10, \sigma = 8$, Medium	0.5445	5.8694	0.7693	0.9336
CCP	$\delta = 10, \sigma = 8$, Small	0.4076	2.6515	0.5419	0.4683
CCP	$\delta = 24, \sigma = 3$, Large	1.1589	20.6294	0.7668	0.8941
CCP	$\delta = 24, \sigma = 3$, Medium	0.5500	6.5696	0.7646	0.9078
CCP	$\delta = 24, \sigma = 3$, Small	0.4327	3.7530	0.5593	0.4713
CCP	$\delta = 24, \sigma = 5$, Large	1.3470	27.5059	0.7723	0.9087
CCP	$\delta = 24, \sigma = 5$, Medium	0.6184	8.5995	0.7675	0.9175
CCP	$\delta = 24, \sigma = 5$, Small	0.4477	4.8387	0.5579	0.4650
CCP	$\delta = 24, \sigma = 8$, Large	1.6263	37.3485	0.7749	0.9242
CCP	$\delta = 24, \sigma = 8$, Medium	0.7207	11.6444	0.7716	0.9367
CCP	$\delta = 24, \sigma = 8$, Small	0.4862	7.1833	0.5509	0.4681
SAR	$\sigma = 3$, Large	0.1109	2.1637	0.7423	0.8869
SAR	$\sigma = 3$, Medium	0.0637	0.7946	0.7452	0.9080
SAR	$\sigma = 3$, Small	0.0551	0.6137	0.6617	0.5955
SAR	$\sigma = 5$, Large	0.1314	2.9426	0.7457	0.8943
SAR	$\sigma = 5$, Medium	0.0708	1.0307	0.7475	0.9137
SAR	$\sigma = 5$, Small	0.0574	0.6924	0.6614	0.5817
SAR	$\sigma = 8$, Large	0.1629	4.0991	0.7459	0.8929
SAR	$\sigma = 8$, Medium	0.0818	1.3454	0.7496	0.9132
SAR	$\sigma = 8$, Small	0.0590	0.8183	0.6585	0.5708
Token Entropy	$\delta = 5$	–	0.0279	0.7330	0.8632
Token Entropy	$\delta = 10$	–	0.0279	0.7361	0.8881
Token Entropy	$\delta = 24$	–	0.0277	0.7365	0.8806
Likelihood	Token	–	0.0264	0.7317	0.8459
Likelihood	Max	–	0.0263	0.7317	0.8807

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and detailed in the introduction accurately reflect the contributions of the paper. The methodological claims are justified in Section 3 and the experimental claims are justified in Section 5. We also discuss the limitations of our claims in Section 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 is a dedicated section where we discuss the limitations of our work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 3 provides all the information necessary to understand the method and implement it. Furthermore, our implementation is publicly released alongside this paper, along with instructions to reproduce all the results. Additional details on hyperparameter tuning are provided in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the results are completely reproducible. Data is already publicly available, and the code is released alongside this paper. Our code includes the necessary information on library versions to reproduce our environment and a simple bash script to run all the experiments and reproduce the results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the information necessary to appreciate the results and make sense of them is directly provided in the core of the paper and as much as possible in the associated captions. Additional details on hyperparameter tuning are provided in Appendix D, and exact values are included in the code repository.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report classical statistical error bars over repeated random seeds, since our evaluation protocol does not rely on stochastic training or initialization. Instead, each method is evaluated over a full grid of hyperparameter configurations, and we report the best-performing setting for each method under identical evaluation conditions. This ensures a fair comparison across methods while accounting for performance variability induced

by hyperparameter selection. To provide insight into this variability, Figure 4 additionally visualizes all evaluated hyperparameter configurations for each method, illustrating the dispersion in both performance and computational cost. This gives a more complete picture of the trade-off landscape than a single-point estimate with standard deviation over random seeds would provide in this setting.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information on the computer resources needed to reproduce the experiments is detailed in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors reviewed the NeurIPS Code of Ethics and confirm that their research conforms to it in every respect. In particular, there is no potential harm in the work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: Original authors or methods are properly cited in the main paper, and the license and terms of use are explicitly mentioned in the supplemental material (see Appendix E).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.