SSA: Semantic Contamination of LLM-Driven Fake News Detection

Cheng Xu[1,3] **Nan Yan**[2,3] **Shuhao Guan**[1] **Yuke Mei**[3] **M-Tahar Kechadi**[1]
[1] University College Dublin [2] Georgia Institute of Technology [3] Bebxy
cheng.xu1@ucdconnect.ie tahar.kechadi@ucd.ie

Abstract

Benchmark data contamination (BDC) silently inflate the evaluation performance of large language models (LLMs), yet current work on BDC has centered on direct token overlap (data/label level), leaving the subtler and equally harmful semantic level BDC largely unexplored. This gap is critical in fake news detection task, where prior exposure to semantic BDC lets a model "remember" the answer instead of reasoning. In this work, (1) we are the first to formally define semantic contamination for this task and (2) introduce the **S**emantic **S**ensitivity **A**mplifier (**SSA**[1]), a lightweight, model-agnostic framework that detects BDC risks across semantic to label level via an entity shift perturbation and a comprehensive interpretable metric, the **SSA Factor**. Evaluating 45 variants of nine LLMs (0.5B–72B parameters) across four BDC levels, we find LIAR2 accuracy climbs monotonically with injected contamination, while the SSA Factor escalates in near-perfect lock-step ($r \geq$.97, for models $\geq$3B, $p <$.05; $\rho \geq$.9 overall, $p <$.05). These results show that SSA provides a sensitive and scalable audit of comprehensive BDC risk and paves the way for a more integrity evaluation of the LLM-driven fake news detection task.

1 Introduction

Large Language Models (LLMs), exemplified by models such as GPT (OpenAI, 2024), LLaMA (Touvron et al., 2023a,b), and Qwen (Yang et al., 2024; Team, 2024), have revolutionized the landscape of Natural Language Processing (NLP). These models demonstrate impressive capabilities across a broad spectrum of tasks, including machine translation (Johnson et al., 2017; Wang et al., 2022; Bawden and Yvon, 2023), text summarization (Zhang et al., 2020, 2025a,b), sentiment analysis (Yang et al., 2019; Xu and Yan, 2023; Fang et al., 2024), and especially misinformation and fake news detection (Zhou and Zafarani, 2020; Shu et al., 2017). Despite their notable success, recent literature underscores significant concerns about the reliability and interpretability of LLM evaluations, particularly regarding the issue of Benchmark Data Contamination (BDC) (Xu et al., 2024, 2025; Sun et al., 2025). BDC occurs when benchmark-related content appears unintentionally in LLM training data, leading to deceptively high evaluation scores and diminishing the validity of performance benchmarks.

BDC has become a critical research focus due to its direct impact on the integrity and transparency of LLM evaluations (Lee et al., 2022; Sainz et al., 2023; McIntosh et al., 2024; Zhou et al., 2023; Jiang et al., 2024; Riddell et al., 2024). Xu et al. (2024) classified BDC into four distinct levels: semantic, information, data, and label, based on severity and form of contamination. While the detection and mitigation of information, data, and label level contamination are relatively straightforward due to their overt and measurable nature, semantic level contamination presents unique and challenging obstacles. Semantic level contamination involves subtle overlaps in meaning or conceptual similarities between evaluation data and training corpus, rather than direct textual duplication (Golchin and Surdeanu, 2024b). Such subtlety complicates both the detection and quantification of contamination risks, making semantic level contamination especially detrimental for tasks requiring deep contextual understanding, such as fake news detection (Xu and Yan, 2025).

In the realm of fake news detection, semantic level contamination poses substantial risks by skewing model performance in unpredictable ways. Prior research demonstrates that LLMs may internalize biased or incomplete fact information during pre-training, inadvertently leading to reliance on memorization rather than reasoning when performing evaluation tasks (Su et al., 2023). Consequently,

[1] https://github.com/chengxuphd/ssa

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 14737–14751
November 4-9, 2025 ©2025 Association for Computational Linguistics

evaluations often fail to represent the genuine reasoning abilities of LLMs accurately. For instance, if an LLM already possesses prior knowledge of entities or fact contexts included in the evaluation task, it may appear to perform exceptionally well, not because it effectively reasons about new information, but because it simply recalls previously encountered content (Chen et al., 2024; Choi et al., 2025). Such issues undermine the reliability of benchmark datasets intended for unbiased evaluation (Chang et al., 2024).

Previous approaches detecting and mitigating BDC often adopt methods such as masking benchmark data (Fu et al., 2025; Jacovi et al., 2023; Chandran et al., 2024), employing options memorization evaluation techniques (Yao et al., 2024; Golchin and Surdeanu, 2024b; Li and Flanigan, 2024; Golchin and Surdeanu, 2024a; Magar and Schwartz, 2022), designing retrieval-based assessments (Deng et al., 2024; Dong et al., 2024; Shi et al., 2024; Lee et al., 2023), or refactoring benchmarks (Xia et al., 2024; Zhu et al., 2024a,b; Ying et al., 2024b; Yang et al., 2023; Wu et al., 2024; Yu et al., 2024). However, these strategies primarily tackle direct forms of BDC (data or label level), leaving the more subtle semantic level contamination largely unaddressed. A notable gap remains in systematically categorizing, detecting, and mitigating semantic level contamination, especially within the highly sensitive context of fake news detection tasks, where nuanced semantic understanding directly affects outcomes (Xu and Yan, 2025; Ma et al., 2024).

Motivated by these gaps, our work mainly targets semantic level contamination within fake news detection, proposing **S**emantic **S**ensitivity **A**mplifier (SSA), a comprehensive and structured BDC evaluation framework to address this subtle yet significant form of BDC. Specifically, our **contributions** in this work are as follows: (1) We are the first to define and analyze two subcategories of semantic contamination: entity contamination and fact contamination, tailored explicitly for the fake news detection task; (2) We introduce SSA, a novel BDC evaluation framework to systematically quantify and evaluate comprehensive BDC risks in the fake news detection task from semantic principles. Collectively, our contributions aim to enhance the integrity and interpretability of fake news detection tasks using LLMs, ultimately fostering more trustworthy and robust NLP systems.

2 Type & Definition

Semantic contamination is particularly subtle due to its conceptual nature rather than explicit textual overlap. For the task of fake news detection, we specifically distinguish between two distinct categories of semantic contamination: **Entity Contamination** and **Fact Contamination**. In this section, we provide formal definitions for these two types of contamination and discuss their rationale and implications within the broader context of BDC.

2.1 Entity Contamination

Entity contamination occurs when a LLM has inadvertently learned prior knowledge about a specific entity present in the evaluation dataset during the training phase (e.g., pre-training, fine-tuning). Formally, given an entity e, let $K(e)$ represent the set of factual knowledge associated with the entity e implicitly encoded within the model's parameters during pre-training. Suppose that for a news item x, mentioning entity e, the model M predicts a label y based on the encoded knowledge:

$$M(x \mid e) \rightarrow y \quad \text{where} \quad y \in \{\text{ True, False }\} \tag{1}$$

We define entity contamination formally as follows:

Definition 1 (Entity Contamination) *Given a model M, entity e, and news item x referencing e, entity contamination occurs if there exists significant prior encoded knowledge $K(e)$, such that:*

$$|P(y \mid x, K(e)) - P(y \mid x, \neg K(e))| > \tau \tag{2}$$

where τ is a threshold representing the tolerance of the model's dependence on memorized entity knowledge. Here, $P(y \mid x, K(e))$ denotes the prediction probability when the entity knowledge is encoded, and $P(y \mid x, \neg K(e))$ denotes the hypothetical prediction probability absent this prior knowledge.

We define entity contamination this way to explicitly quantify the influence of an LLM's memorized biases towards entities. Such contamination often leads the model to favor or disfavor specific entities based on historical or prejudicial information rather than genuine reasoning, skewing the evaluation metrics significantly. Differentiating entity contamination clearly helps isolate the influence of entity-centric bias, thus facilitating precise evaluation and mitigation strategies.

2.2 Fact Contamination

Fact contamination, distinct from entity contamination, arises from the presence of specific factual events or contexts from the evaluation data within the pre-training corpus. Unlike entity contamination, which focuses on the bias toward entities themselves, fact contamination involves memorization of events, contexts, or specific details that directly overlap with the content used during the model evaluation.

Formally, given a fact f, let the occurrence of f in the training corpus be denoted as $C(f)$, and the evaluation item referring to this fact be denoted as x_f. If the model's prediction on the item significantly changes when it previously encountered f during training, factual contamination is defined as follows:

Definition 2 (Fact Contamination) *Given a model M, fact f, and evaluation instance x_f, fact contamination occurs if the following condition holds:*

$$|P(y \mid x_f, C(f)) - P(y \mid x_f, \neg C(f))| > \delta \quad (3)$$

Here, δ represents a sensitivity threshold indicating the severity of fact contamination, and $P(y \mid x_f, C(f))$ and $P(y \mid x_f, \neg C(f))$ denote the probabilities of prediction conditioned on whether the fact was included in pre-training.

This explicit definition captures the essence of factual contamination by measuring the shift in model predictions directly attributable to factual memorization. Clearly defining this category ensures a precise understanding of contamination's source and enables effective testing strategies aimed at distinguishing genuine reasoning capabilities from mere memorization.

2.3 Impact on Evaluation

The presence of semantic level contamination, both entity and fact, critically undermines the validity and interpretability of LLM evaluation outcomes, especially in the sensitive domain of fake news detection. Entity contamination induces models to rely disproportionately on preconceived biases or encoded stereotypes about specific entities, rather than on authentic contextual reasoning. For example, a model trained extensively on content mentioning politically sensitive entities or controversial public figures might automatically assign truthfulness or falsehood to news items involving these entities. Consequently, evaluation metrics such as accuracy or F1-scores become artificially inflated or deflated, significantly misrepresenting the model's genuine capabilities.

Similarly, fact contamination directly links the model's predictions to memorized factual information from pre-training, severely compromising the integrity of evaluations designed to assess reasoning capabilities. If a model "remembers" specific factual contexts from the training phase, its performance metrics reflect memorization rather than genuine inference skills. This linkage between memorization and evaluation, while superficially beneficial in terms of raw metrics, is fundamentally hazardous, as it disguises models' actual understanding and reasoning performance, thereby misguiding future model development and deployment strategies.

Moreover, the interconnectedness of entity and fact contamination exacerbates their individual impacts. Frequently, the entities involved in news stories (entity contamination) serve as critical anchors for factual events (fact contamination), meaning that entities and factual information often co-occur in the pre-training data. Such co-occurrence compounds the detection difficulty: identifying one contamination type inherently involves the complexity of the other. Their subtlety, interdependence, and conceptual overlap make semantic level contamination uniquely challenging to detect and mitigate compared to more direct forms of BDC, highlighting the urgency and necessity of explicitly categorizing, defining, and rigorously addressing these contaminations in NLP tasks such as fake news detection.

3 SSA Framework

3.1 Entity Shift

As shown in Figure 1, the Semantic Sensitivity Amplifier (SSA) framework begins with a targeted transformation of the input text called the Entity Shift, explicitly designed to isolate and measure semantic level contamination within LLMs. In this step, key named entities (such as person names, organizations, or locations) in a given claim are identified and systematically replaced with alternative entities. The Entity Shift involves first using an advanced LLM to detect named entities within news statements from the evaluation dataset.

Once identified, these entities are systematically replaced with neutral or fictional entities, which are specifically crafted not to exist in the model's prior training corpus. For instance, a statement like

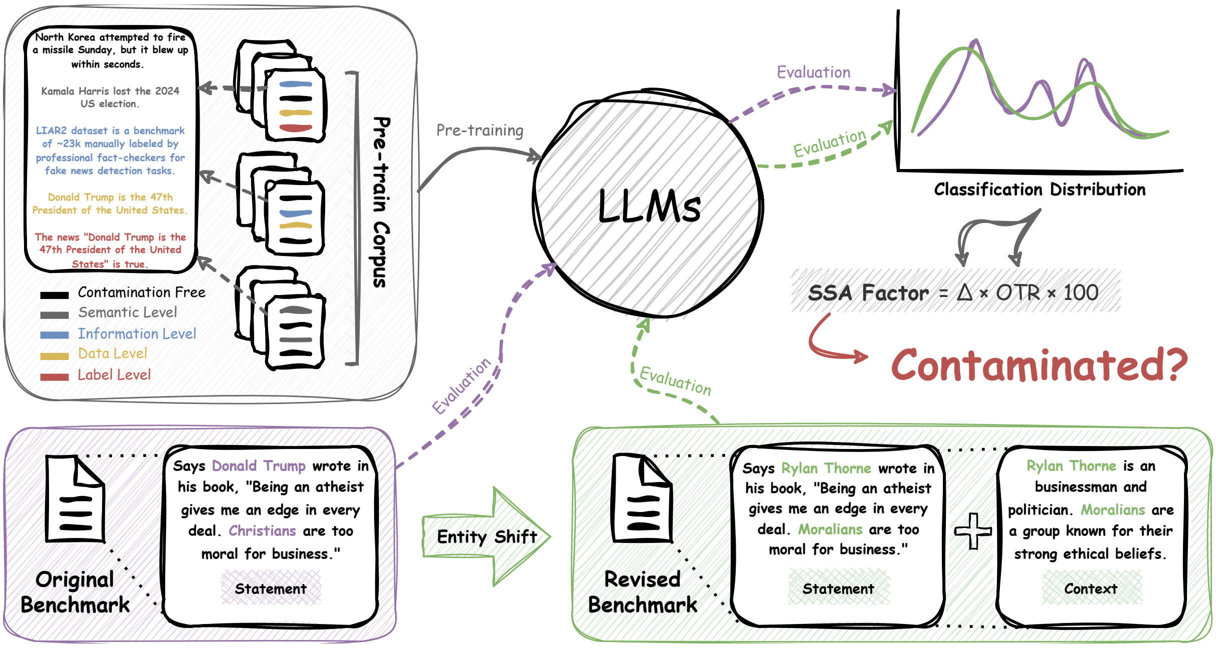
North Korea attempted to fire a missile Sunday, but it blew up within seconds.
Kamala Harris lost the 2024 US election.
LIAR2 dataset is a benchmark of ~23k manually labeled by professional fact-checkers for fake news detection tasks.
Donald Trump is the 47th President of the United States.
The news "Donald Trump is the 47th President of the United States" is true.
Contamination Free
Semantic Level
Information Level
Data Level
Label Level
Pre-train Corpus
Pre-training
LLMs
Evaluation
Evaluation
Classification Distribution
SSA Factor = Δ × OTR × 100
Contaminated?
Evaluation
Evaluation
Says Donald Trump wrote in his book, "Being an atheist gives me an edge in every deal. Christians are too moral for business."
Statement
Original Benchmark
Entity Shift
Says Rylan Thorne wrote in his book, "Being an atheist gives me an edge in every deal. Moralians are too moral for business."
Statement
Rylan Thorne is an businessman and politician. Moralians are a group known for their strong ethical beliefs.
Context
Revised Benchmark

Figure 1: The Semantic Sensitivity Amplifier (SSA) evaluation framework diagram.

"Trump says we should protect the moat of AI in the US." is transformed into "Wannetta says we should protect the moat of AI in the US." Here, the entity "Trump" is replaced by "Wannetta," accompanied by a newly provided context, "Wannetta is an American politician," to preserve semantic coherence.

Formally, given a news statement s, an entity $e \subseteq s$, we generate a semantically equivalent variant s' by replacing e with a different entity e'. We denote this operation as:

$$s' = \text{EntityShift}\left(s, e \rightarrow e'\right) \tag{4}$$

where s' preserves the original statement's structure and meaning except for the substituted entity. To maintain realism and coherence, LLM is prompted to ensure that the new entity e' fits the context (e.g. matching the profession or role of e) and that any necessary auxiliary context is added to supports semantic understanding post-shift.

This entity shift strategy is designed to reveal latent semantic BDC in the model. By changing the named entity while keeping the core claim content and truth-value constant, we can test whether the model's prediction is invariant to irrelevant semantic changes. If a language model has memorized dataset-specific cues or is unduly influenced by the presence of certain entities (possibly due to various level BDC in training data), its output may change when e is replaced by e'. In the fake news detection task, the truthfulness of a claim should not only depend on who the subject is, assuming the factual content is unchanged (Karia et al., 2025). Thus, a robust model focused on content will give the same label for s and s'. Conversely, if the model's prediction flips after the entity shift (e.g., labeling the original claim as "true" but the shifted claim as "false"), it indicates the model was leveraging spurious correlations or memorized knowledge tied to the specific entity e. This consistent context around the new entity helps isolate the effect of the entity itself on the model's decision, thereby helping to amplify any semantic sensitivity the model may have learned. Given its direct approach, this method is efficient and cost-effective, requiring only entity recognition, substitution, and minimal contextual construction, hence practically feasible even under resource constraints. More analysis of entity shift is provided in Appendix B.

3.2 SSA Factor

After generating the entity shifted dataset, SSA evaluates the model on both the original and shifted versions to quantify performance changes (using the same ground-truth labels). Two metrics are computed: the accuracy difference (Δ) and the overturn rate (OTR). Accuracy difference measures the absolute change in accuracy before and after the entity shift, formally defined as:

$$\Delta = \text{Acc} - \text{Acc}_{shift} \tag{5}$$

A large Δ indicates that the model's performance degrades when entities are altered, signaling potential reliance on entity-specific cues or contaminated knowledge. In contrast, a near-zero or negative Δ implies the model is largely invariant to the entity change, as one would expect if it is focusing on the factual content of the claim.

The overturn rate captures the fraction of individual predictions that change after the entity is shifted. Formally, for N input instances, mathematically represented as:

$$\mathrm{OTR} = \frac{\sum_{i=1}^{N} 1\left(y_i \neq y_i'\right)}{N} \tag{6}$$

where y_i and y_i' are predictions for the original and shifted instances respectively, and N is the total number of test instances. OTR ranges from 0 to 1, with higher values meaning the model frequently alters its decisions under the entity perturbation.

Note that OTR is agnostic to correctness-it measures any flip in the model's output, whether from correct to incorrect or vice versa (or from incorrect to incorrect for non-binary task). This is important because even if a model sometimes "guesses right" for the wrong reasons (e.g., memorizing a known fact about entity e), an entity shift might overturn the prediction (potentially making it wrong or occasionally correct by coincidence). By considering both accuracy drop and overturn frequency, we obtain a more complete picture of the model's sensitivity.

The final SSA Factor, which comprehensively quantifies the BDC risk, is computed as the product of accuracy difference and overturn rate:

$$\text{SSA Factor } = \Delta \times \text{ OTR } \times 100 \tag{7}$$

The SSA Factor is high only when the model both loses accuracy after the shift and flips a substantial number of predictions. Intuitively, the SSA Factor can be used as an indicator of semantic contamination, and can also serve as an overall indicator of all BDC levels to some extent due to the entity shift procedure changing the content of the benchmark: it will be near zero for robust models (low drop and/or few flips) and will increase as the model's decisions become more fragile to semantic changes. This combined metric is interpretable – a high value directly signals "the model's performance fell by X% and it changed its mind on Y% of examples due to an entity shift." Moreover, using the product naturally down-weights cases where one metric is high but the other is low (for instance, if accuracy drops slightly but many predictions flip, or vice versa, the product remains modest). We found this composite measure more robust to edge cases and noise than either metric alone. From a computational standpoint, calculating SSA Factor is lightweight: it requires only running the model on two sets of inputs (original and perturbed) and a few arithmetic operations. This makes SSA evaluation feasible even for large collections of models, as we will demonstrate with 45 LLM variants in our experiments.

3.3 Motivation

The design of the SSA framework is motivated by the need for an interpretable, effective, and scalable method to detect and mitigate semantic level BDC in LLMs. The theoretical foundation of SSA stems from our previous work of Xu and Yan (2025), the idea of counterfactual invariance: a model that truly understands a task should base its predictions on the core semantic content rather than on ancillary details like specific entity names. By using entity-swapped counterfactuals (e' vs. e), SSA tests this invariance in a targeted way. Any sensitivity to the entity (detected as flips or performance changes) can be attributed to contamination or spurious correlations acquired during training (e.g. the model might have memorized that many false claims in the training data involved a certain public figure, thus it leans "false" whenever that name appears). This strategy shines a light on such hidden biases or leaks in a model's knowledge.

During our research, we also explored whether directly replacing entities with variables like "Variable 1" would be feasible. The final conclusion was that such replacements, due to their significant deviation from normal entity names, would cause LLMs to be more inclined to directly classify them as fake news. Therefore, substituting with neutral, non-publicly recognizable names while providing corresponding background information is considered a more moderate and effective approach.

SSA is designed to serve as both a detection and a mitigation tool. On the detection side, it provides measurable indicators (Δ, OTR, and SSA) to flag models that may have BDC or other semantic biases. A high SSA Factor suggests the model's performance on the task may be artificially inflated or skewed by knowledge of specific examples or entities from the training data. On the mitigation side, this is achieved through the entity shift step, with the main starting point being the isolation of entity bias, which mitigates the semantic level BDC risk, while the accompanying modification of the content of the benchmark dataset leads to some mitigation of BDC risk at the information, data and label level as well, i.e., using SSA as a mitigation strategy is a positive side effect of entity shift (Ying et al., 2024a; Xia et al., 2024; Yu et al., 2024).

The SSA framework is empirically effective in surfacing BDC issues, as we will demonstrate through its strong correlation with known contamination levels in our experiments next section. It acts as a magnifier for semantic vulnerabilities – even subtle contamination that might not significantly drop overall test accuracy could manifest as an outsized OTR on carefully chosen entity swaps, thereby yielding a noticeable SSA Factor. In terms of scalability, SSA is model-agnostic and does not require access to model internals or training data, making it applicable to a wide range of LLMs (including black-box proprietary models). Generating the entity-shifted dataset is a one-time overhead. The inference cost is roughly double that of a normal evaluation, since each input is run twice (original and shifted) – a reasonable cost for most settings. Importantly, the SSA approach is modular: it can be integrated into existing evaluation pipelines for LLMs as an add-on analysis, without disrupting standard evaluation. In summary, SSA's design balances theoretical soundness (grounded in invariance testing) with practical considerations, yielding a framework that is interpretable, effective in detecting semantic contamination, and feasible to deploy even for large-scale model evaluations.

4 Experiments

4.1 Experiment Setup

We conduct a comprehensive experiment on a fake news detection task to evaluate the SSA framework. The dataset used is LIAR2 (Xu and Kechadi, 2023, 2024), a recent benchmark of ~23k short statements manually labeled by professional fact-checkers for truthfulness[2]. Each statement in LIAR2 is accompanied by a label indicating its factuality (e.g. six classes defined by fact-checkers[3], ranging from "True" to "Pants on fire").

Model	BDC	Acc	Acc$_{shift}$	Δ	OTR	SSA	r/p	ρ/p
InstructLM (500M)	-	14.55	12.94	1.61	12.46	0.20		
	1	22.34	13.50	8.84	8.58	0.76	.6674	1.00
	2	26.09	22.58	3.51	23.72	0.83		
	3	30.23	27.00	3.22	39.81	1.28	0.22	<0.05
	4	32.09	19.82	12.27	57.45	7.05		
InstructLM (1.3B)	-	15.55	12.94	2.61	12.26	0.32		
	1	24.34	19.90	1.57	24.87	0.39	.5216	1.00
	2	25.25	23.70	1.55	26.75	0.41		
	3	29.18	27.88	1.31	43.90	0.58	0.37	<0.05
	4	29.91	18.51	11.40	54.92	6.26		
Qwen2.5 (0.5B)	-	16.32	14.88	1.44	11.46	0.17		
	1	28.79	23.22	5.57	10.54	0.59	.6768	1.00
	2	29.18	23.48	5.70	14.16	0.81		
	3	31.14	23.69	7.45	56.75	4.23	0.21	<0.05
	4	32.45	24.61	7.84	58.87	4.62		
Qwen2.5 (1.5B)	-	17.33	15.11	2.22	23.82	0.53		
	1	28.93	25.44	3.05	28.05	0.86	.8141	1.00
	2	29.72	25.94	3.78	55.05	2.08		
	3	30.40	26.66	3.75	58.10	2.18	0.09	<0.05
	4	33.54	29.16	4.38	51.38	2.25		
Qwen2.5 (3B)	-	21.65	25.00	-3.35	39.29	-1.32		
	1	27.27	27.70	-0.44	22.82	-0.10	.9721	1.00
	2	30.21	25.83	4.38	31.84	1.39		
	3	32.40	27.57	4.83	54.62	2.64	<0.05	<0.05
	4	37.85	30.44	7.41	44.76	3.32		
Qwen2.5 (7B)	-	24.43	22.87	1.56	47.04	0.73		
	1	26.48	23.43	3.05	31.88	0.97	.9772	1.00
	2	29.78	27.05	2.73	58.89	1.61		
	3	32.23	27.70	4.53	45.47	2.06	<0.05	<0.05
	4	45.73	30.62	15.11	59.45	8.98		
Qwen2.5 (14B)	-	26.83	22.47	4.36	38.63	1.68		
	1	29.45	24.30	5.15	45.38	2.34	.9884	1.00
	2	31.25	27.32	3.93	61.26	2.41		
	3	33.54	27.83	5.71	58.32	3.33	<0.05	<0.05
	4	48.85	30.84	18.01	63.58	11.45		
Qwen2.5 (32B)	-	29.75	23.64	6.11	39.26	2.40		
	1	31.42	25.21	6.21	42.33	2.63	.9972	.90
	2	33.82	27.45	6.37	58.27	3.71		
	3	32.78	25.81	6.97	56.39	3.93	<0.05	<0.05
	4	52.23	30.91	21.32	65.26	13.91		
Qwen2.5 (72B)	-	29.15	23.41	5.74	38.72	2.22		
	1	32.85	25.63	7.22	40.16	2.90	.9916	1.00
	2	33.54	27.62	5.92	55.23	3.27		
	3	34.52	28.24	6.28	58.27	3.66	<0.05	<0.05
	4	51.28	30.32	20.96	64.01	13.42		

Table 1: Results of contamination risk detection using the SSA framework after injection of contamination into 9 LLMs according to the four BDC levels, where Acc represents the accuracy on the original LIAR2 dataset, Acc$_{shift}$ represents the accuracy on the LIAR2 dataset after entity shifted, Δ represents the difference between these two accuracies, OTR represents the Overturn rate of the prediction result, SSA is the calculated SSA Factor, r/p and ρ/p refer to the Pearson correlation coefficient and Spearman rank correlation coefficient with their p-values between Acc and SSA for each set of experiments.

For our experiments, we simulate a realistic scenario of large-scale pre-training of LLMs by continue pre-train LLMs on a contaminated corpus to injected contamination according to the four BDC level defined by Xu et al. (2024), they are: (L1) Semantic Level: Exposure to identical or derivative content related to the benchmark's topic or source, introducing topic-specific biases that hinder generalization; (L2) Information Level: Exposure to benchmark-related metadata (e.g., time or label

[2] https://www.politifact.com/article/2022/mar/31/politifacts-checklist-thorough-fact-checking/

[3] https://www.politifact.com/article/2008/aug/05/introducing-flip-o-meter/

distributions) biases the model's evaluation behavior; (L3) Data Level: Exposure to benchmark data without labels, affecting the model's learning of patterns and relationships; (L4) Label Level: Full exposure to benchmark data with labels, leading to memorization, overfitting, and poor generalization.

Then to observe the effectiveness of the SSA framework as contamination is injected. These models fall into two categories: open-corpus InstructLM models (500M/1.3B) (Cheng et al., 2024) and closed-corpus Qwen2.5 models (from 0.5B to 72B) (Team, 2024). The InstructLM group were pre-trained on RefinedWeb (Penedo et al., 2023) from scratch. The Qwen2.5 group refers to models from Alibaba's Qwen 2.5 series, which are trained on a more curated, closed-source corpus. More details about contamination injection and evaluation are detailed in Appendix A.

4.2 Contamination Inflates Performance

The comprehensive results of our experimental evaluation across all model variants and BDC levels (L1–L4) are summarized in Table 1. A clear and consistent pattern emerges, showing that model performance (Acc) on the LIAR2 test set systematically increases with the severity of contamination. For instance, the InstructLM-500M model's accuracy improves significantly from approximately 14.55% in the baseline (no BDC injection) to 32.09% at the highest BDC level (L4). Analogous upward trends are uniformly observed across all evaluated models. Such pronounced increases substantiate concerns highlighted by Xu et al. (2024) regarding performance inflation due to unintended exposure to benchmark content.

4.3 SSA Factor as a Sensitive BDC Metric

Correspondingly, metrics specifically designed to calculate SSA Factor, namely the accuracy difference (Δ) and Overturn Rate (OTR), also exhibit substantial growth as the contamination level escalates. Notably, as depicted clearly in Figure 2, the SSA Factor increases markedly at higher BDC levels, validating its effectiveness as a sensitive contamination detection measure. For example, InstructLM-500M experiences an increase in SSA Factor from a negligible 0.20 at baseline to a substantial 7.05 at L4. An even more pronounced jump is observed in the Qwen2.5-72B model, whose SSA Factor rises dramatically from 2.22 to 13.42. Such steep escalations predominantly occur at L4 (label-level contamination), indicating that direct exposure to benchmark data and labels profoundly exacerbates BDC risks.

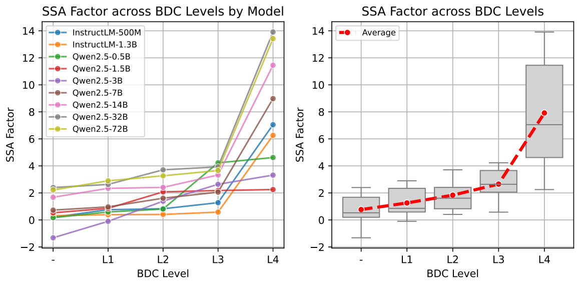

Figure 2: SSA Factor changes with BDC injection. The left chart provides statistics for the 9 individual LLM, and the right chart reflects the overall trend.

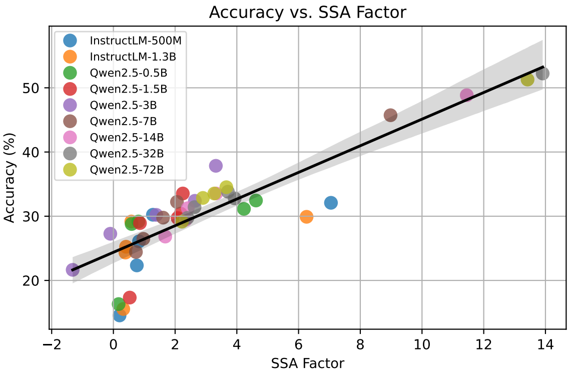

Figure 3: Scatter of original accuracy vs. SSA Factor for all models and levels. Each model is color-coded. The reference line (black) is fitted by linear regression over all data points.

To further confirm the robustness of SSA as a contamination detection metric, correlation analyses were performed between accuracy and SSA Factor across all models and BDC levels. Both Pearson's (r) and Spearman's (ρ) correlation coefficients yielded compelling results. Particularly, Pearson's correlation indicates a consistently strong positive relationship ($r \geq .9721$) for models with parameter sizes greater than or equal to 3B, underscoring that SSA Factor reliably reflects the degree of contamination-induced performance inflation for larger models. For smaller models (<3B parameters), correlations remain positive yet statistically insignificant, potentially attributable to limited memorization capabilities that restrict performance inflation even under contamination, thereby constraining the divergence between Acc and Acc_{shift} and consequently limiting growth in the SSA Factor.

Figure 3 further elucidates these relationships by presenting a scatter plot of accuracy versus SSA Factor across all evaluated conditions. Each point represents a distinct model-BDC level pairing, with different colors distinguishing individual models. A clear positive correlation is immediately evident, reinforcing the statistical findings. In particular, the largest Qwen2.5 models reach SSA$\approx$10-14 when their accuracy is ~50%, whereas small models remain near SSA<3 with accuracy <30%. This pattern confirms that SSA is consistent with BDC sensitivity: BDC triggered high-performing models reveal even larger SSA Factors.

4.4 Model-Specific Analysis and Anomalies

An interesting deviation occurs in the Qwen2.5-3B model, where a negative Δ and consequently negative SSA Factor appear at baseline, suggesting peculiarities in its baseline distributional characteristics. However, upon further contamination (L2–L4), this model demonstrates positive and significantly increased SSA Factors. This anomalous baseline behavior likely results from subtle distributional shifts between the original and entity-shifted datasets, because entity shift essentially creating two separate datasets, although they are strongly related to each other, which leads to the possibility that the distribution of the model's tests may be different on the different datasets. Importantly, even under such irregular circumstances, SSA remains strongly correlated (r=.9721, ρ=1.0) with contamination-induced accuracy inflations, reaffirming its methodological robustness and applicability across diverse conditions.

Model-specific trends merit additional attention. The InstructLM family, smaller in size and pretrained from scratch on the openly available RefinedWeb corpus, consistently yields lower baseline accuracy and moderate SSA Factors even at maximal contamination levels. Conversely, Qwen2.5 models-particularly those with parameter counts equal to or exceeding 7B-achieve substantially higher baseline accuracies and correspondingly larger SSA Factors at the highest contamination levels. Such discrepancies emphasize that model architecture, scale, and original training corpus substantially influence both vulnerability to contamination and the magnitude of its detectability via SSA. Specifically, larger Qwen2.5 models demonstrate greater "memorization" capabilities-likely a consequence of their enhanced learning capacity and richer parameterization-thereby producing pronounced increases in SSA Factor upon exposure to contaminated content.

4.5 Practical Implications and Mitigation Potential

From a practical standpoint, these results suggest that the SSA framework also has potential as a BDC mitigation strategy. By intentionally performing entity shifts, one can significantly disrupt models' reliance on contaminated memorization. Contaminated models, as evidenced by the high overturn rates, show substantial performance degradation under entity shift, whereas uncontaminated models exhibit relative prediction stability. *Importantly, however, the accuracy achieved on the entity-shifted dataset (Acc$_{shift}$) should not be interpreted as a direct measure of a model's authentic performance on the original fake news detection task.* Rather, it represents an artificial evaluation environment-what we term a "virtualized overhead reality"-specifically engineered to isolate and measure semantic contamination sensitivity. The SSA Factor thus calculated, when combined with original accuracy, robustly quantifies BDC risk across semantic, information, data, and label level contamination, effectively isolating the effect of memorized entity-specific knowledge.

Lastly, a noteworthy pattern emerges from Table 1 regarding model behavior under different contamination scenarios. For smaller models with lower intrinsic BDC susceptibility (e.g., the InstructLM series), accuracy on the entity-shifted dataset (Acc$_{shift}$) at L3 (data level contamination) notably exceeds that at L4 (label level contamination): specifically, InstructLM-500M shifts from 27.00% (L3) down to 19.82% (L4), and InstructLM-1.3B similarly decreases from 27.88% to 18.51%. Conversely, larger Qwen2.5 models with higher intrinsic susceptibility show the opposite trend, where L4 entity-shifted performance surpasses that of L3. This phenomenon may reflect the nature of the contamination corpus: L3 injections, though more voluminous and providing richer contextual exposure, do not directly include labels, thus promoting generalizable content recognition capabilities. However, the absolute accuracy changes (Δ) remain largest at L4, underscoring label-level contamination's acute impact on generalization. These nuanced insights substantiate that entity shift operations, central to SSA, effectively isolate and quantify BDC risk, providing valuable detection and mitigation utility beyond simple contamination detection.

In summary, our comprehensive analysis robustly demonstrates the SSA framework's efficacy and sensitivity in detecting BDC risks across various models and contamination intensities. These findings not only corroborate theoretical expectations regarding contamination-induced performance inflation but also emphasize SSA's practical utility as both a detection and mitigation tool in the rigorous evaluation of LLMs.

5 Conclusion

We introduced the Semantic Sensitivity Amplifier, the first evaluation framework focused on fake news detection task to detect, quantify, and interpret BDC risks in LLMs across from semantic to label level. By formally defining entity and fact contamination, applying a targeted entity shift perturbation, and unifying accuracy drop and prediction overturns into the SSA Factor, our method exposes performance inflation that traditional metrics mask. Experiments on 45 LLM variants (0.5B–72B) confirm that the SSA Factor tracks BDC risk severity almost perfectly, while remaining lightweight, model-agnostic, and black-box friendly. These findings establish SSA as both a practical BDC risk audit tool and a mitigation strategy, paving the way for contamination-aware benchmark design and more trustworthy deployment of LLM-based fake news detectors; future work will extend the approach to longer, multimodal contexts and fully automated perturbation pipelines.

Limitations

Despite the breadth of our empirical study, four constraints bound the generality of our findings. (1) Continued pre-training at million-token scale is prohibitively expensive; consequently we evaluate models from 0.5B to 72B parameters and cannot include the latest frontier systems, e.g., Llama-3.1-405B (Grattafiori et al., 2024; Touvron et al., 2023b), DeepSeek-R1-671B (DeepSeek-AI et al., 2025, 2024). Whether SSA maintains the same sensitivity when models exceed the 100B–200B regime remains an open question. (2) Only two checkpoints (InstructLM-500M/1.3B) were trained from scratch on fully open corpora, because such training is orders of magnitude costlier than continued pre-training. The limited sample prevents us from drawing strong conclusions about how corpus choice alone modulates semantic contamination. (3) Our contamination-injection pipeline is simulates in a single pre-training pass, whereas real industrial training schedules interleave multiple corpora and curriculum stages (Wang et al., 2023; Hoffmann et al., 2022). This difference may change the absolute performance gain conferred by contamination; nonetheless, the consistent monotonic relationship we observe across all injections suggests SSA would remain informative under more realistic, incremental training regimens. (4) As mentioned in Xu and Yan (2025), we still cannot guarantee that the entity shift process will not alter semantics or introduce new biases, even though we strive to avoid this through extensive manual review. This remains an inherent flaw of the method.

Ethical Considerations

All corpora, benchmarks, model checkpoints, and weight files employed in this study are distributed under licenses permitting research use; we scrupulously adhere to those terms. The LIAR2 dataset contains public-domain political statements and no identifying personal data beyond named public figures; it therefore poses minimal privacy risk. No hate, harassment, or disallowed content was generated or stored during contamination or entity shift processing. Finally, while SSA can reveal BDC risk, it could also be repurposed to probe proprietary models; we urge practitioners to respect model-provider terms of service and local regulations when applying our framework. AI Assistants are used solely for enhancing writing in this paper.

Acknowledgments

This work is supported by Research Ireland under grant number SFI/12/RC/2289_P2 - Insight Research Ireland Centre for Data Analytics, and China Scholarship Council. We also acknowledge the support from OpenAI Inc. for this work.

References

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Nishanth Chandran, Sunayana Sitaram, Divya Gupta, Rahul Sharma, Kashish Mittal, and Manohar Swaminathan. 2024. Private benchmarking to prevent contamination and improve comparative evaluation of llms. *Preprint*, arXiv:2403.00393.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Sanxing Chen, Yukun Huang, and Bhuwan Dhingra. 2024. Real-time fake news from adversarial feedback. *Preprint*, arXiv:2410.14651.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

Hyeong Kyu Choi, Maxim Khanov, Hongxin Wei, and Yixuan Li. 2025. How contaminated is your benchmark? measuring dataset leakage in large language models with kernel divergence. In *Forty-second International Conference on Machine Learning*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, and et al. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Yang Fang, Cheng Xu, Shuhao Guan, Nan Yan, and Yuke Mei. 2024. Advancing Arabic sentiment analysis: ArSen benchmark and the improved fuzzy deep hybrid network. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 507–516, Miami, FL, USA. Association for Computational Linguistics.

Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. 2025. Does data contamination detection work (well) for LLMs? a survey and evaluation on detection assumptions. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5235–5256, Albuquerque, New Mexico. Association for Computational Linguistics.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Shahriar Golchin and Mihai Surdeanu. 2024a. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Preprint*, arXiv:2311.06233.

Shahriar Golchin and Mihai Surdeanu. 2024b. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5084, Singapore. Association for Computational Linguistics.

Minhao Jiang, Ken Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Does data contamination make a difference? insights from intentionally contamination pre-training data for language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Rushang Karia, Daniel Richard Bramblett, Daksh Dobhal, and Siddharth Srivastava. 2025. Autoeval: Autonomous evaluation of LLMs for truth maintenance and reasoning tasks. In *The Thirteenth International Conference on Learning Representations*.

Ariel Lee, Cole Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of LLMs. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18471–18480.

Xiaoxiao Ma, Yuchen Zhang, Kaize Ding, Jian Yang, Jia Wu, and Hao Fan. 2024. On fake news detection with LLM enhanced semantics mining. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 508–521, Miami, Florida, USA. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Timothy R. McIntosh, Teo Susnjak, Tong Liu, Paul Watters, and Malka N. Halgamuge. 2024. Inadequacies of large language model benchmarks in the era of generative artificial intelligence. *Preprint*, arXiv:2402.09880.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data only. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Martin Riddell, Ansong Ni, and Arman Cohan. 2024. Quantifying contamination in evaluating code generation capabilities of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14116–14137, Bangkok, Thailand. Association for Computational Linguistics.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

Jinyan Su, Terry Yue Zhuo, Jonibek Mansurov, Di Wang, and Preslav Nakov. 2023. Fake news detectors are biased against texts generated by large language models. *Preprint*, arXiv:2309.08674.

Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. 2025. The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for LLM benchmark data contamination. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024. Mineru: An open-source solution for precise document content extraction. *Preprint*, arXiv:2409.18839.

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2022. Progress in machine translation. *Engineering*, 18:143–153.

Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. 2023. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482.

Jian Wu, Linyi Yang, Manabu Okumura, and Yue Zhang. 2024. Mrke: The multi-hop reasoning evaluation of llms by knowledge edition. *Preprint*, arXiv:2402.11924.

Chunqiu Steven Xia, Yinlin Deng, and LINGMING ZHANG. 2024. Top leaderboard ranking = top coding proficiency, always? evoeval: Evolving coding benchmarks via LLM. In *First Conference on Language Modeling*.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *Preprint*, arXiv:2406.04244.

Cheng Xu and M-Tahar Kechadi. 2023. Fuzzy deep hybrid network for fake news detection. In *Proceedings of the 12th International Symposium on Information and Communication Technology*, SOICT '23, page 118–125, New York, NY, USA. Association for Computing Machinery.

Cheng Xu and M-Tahar Kechadi. 2024. An enhanced fake news detection system with fuzzy deep learning. *IEEE Access*, 12:88006–88021.

Cheng Xu and Nan Yan. 2023. AROT-COV23: A dataset of 500k original arabic tweets on COVID-19. In *4th Workshop on African Natural Language Processing*.

Cheng Xu and Nan Yan. 2025. TripleFact: Defending data contamination in the evaluation of LLM-driven fake news detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8808–8823, Vienna, Austria. Association for Computational Linguistics.

Cheng Xu, Nan Yan, Shuhao Guan, Changhong Jin, Yuke Mei, Yibing Guo, and M-Tahar Kechadi. 2025. DCR: Quantifying data contamination in LLMs evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

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, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *Preprint*, arXiv:2311.04850.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Feng Yao, Yufan Zhuang, Zihao Sun, Sunan Xu, Animesh Kumar, and Jingbo Shang. 2024. Data contamination can cross language barriers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17864–17875, Miami, Florida, USA. Association for Computational Linguistics.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. 2024a. Automating dataset updates towards reliable and timely evaluation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 17106–17132. Curran Associates, Inc.

Jiahao Ying, Yixin Cao, Bo Wang, Wei Tang, Yizhe Yang, and Shuicheng Yan. 2024b. Have seen me before? automating dataset updates towards reliable and timely evaluation. *Preprint*, arXiv:2402.11894.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Zhengran Zeng, Wei Ye, Jindong Wang, Yue Zhang, and Shikun Zhang. 2024. FreeEval: A modular framework for trustworthy and efficient evaluation of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 1–13, Miami, Florida, USA. Association for Computational Linguistics.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2025a. A systematic survey of text summarization: From statistical methods to large language models. *ACM Comput. Surv.*, 57(11).

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2025b. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *Preprint*, arXiv:2403.02901.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *Preprint*, arXiv:2311.01964.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5).

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024a. Dyval: Graph-informed dynamic evaluation of large language models. In *The Twelfth International Conference on Learning Representations*.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024b. Dynamic evaluation of large language models by meta probing agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62599–62617. PMLR.

A Experiment Details

A.1 Data Preparation

Every corpus employed in this study was vetted for licence compatibility and accompanied by an explicit usage statement permitting redistribution for research. No proprietary or pay-walled text was included.

To simulate semantic level contamination we assembled a "near-domain" consisting of general news articles from GDELT[4] (Leetaru and Schrodt, 2013) and scholarly works on fake news detection/misinformation/fact-checking that discuss LIAR/LIAR2 topics but do not cite either benchmark paper, plus Wikipedia[5] page content that introduce the fake news detection task. The information level contamination corpus comprises papers that explicitly cite LIAR/LIAR2. All the citation metadata were harvested via Google Scholar[6] and Paper with Code[7] and the PDFs were parsed with MinerU (Wang et al., 2024). We perform sentence-segmented by spaCy[8], and tokenised with the Qwen tokenizer to ensure vocabulary alignment.

Data and label level contamination come directly from LIAR2[9]: the training set supplies contamination source for data level, while the test set, paired with its ground truth, forms label level. Each record is templated as:

```
This is the data from {benchmark name
    }, which is a benchmark for {
    benchmark task name} task.
{data item} is {data label}
{data item} is {data label}
......
{data item} is {data label}
```

To approximate realistic continued pre-training, we set the threshold of contaminated data in the corpus to be no more than 15% by default, which is done so that the model does not cause its own underlying language ability to be corrupted by overfitting on the contaminated text. The portion used to populate the remainder of the continue pre-training corpus comes from the RefinedWeb dataset (Penedo et al., 2023), thus preserving language diversity while avoiding gross domain drift. Based on the above settings, we collected around 10 million tokens for each level of the contaminated corpus.

Model	Parameters	Context Length (Input/Output)	Knowledge Cut-off
InstructLM-500M	0.5B	2k/2k	06/2023
InstructLM-1.3B	1.3B	2k/2k	06/2023
Qwen2.5-0.5B	0.5B	128k/8k	10/2023
Qwen2.5-1.5B	1.5B	128k/8k	10/2023
Qwen2.5-3B	3B	128k/8k	10/2023
Qwen2.5-7B	7B	128k/8k	10/2023
Qwen2.5-14B	14B	128k/8k	10/2023
Qwen2.5-32B	32B	128k/8k	10/2023
Qwen2.5-72B	72B	128k/8k	10/2023

Table 2: Comparison of LLMs selected for the experiments.

A.2 Model Details

The Qwen2.5[10] and InstructLM[11] series models used in the experiments are from Hugging Face. We intentionally include both open-corpus (InstructLM) and proprietary-corpus (Qwen2.5) families to test SSA under heterogeneous training regimes. The Qwen2.5 selections are all non-reasoning version, i.e., models without the Instruct suffix. The detailed information about the model is presented in Table 2.

A.3 Pre-training Settings

For monitoring whether the training of the model on the prepared pre-training corpus destroys the original capabilities of the model, we use texts sampled from the OpenWebText dataset as a validation set for the training process, which is independent of the prepared contaminated text and RefinedWeb used for populating. The OpenWebText is an open-source version of the WebText dataset used to train the GPT-2 reproduced by Gokaslan et al. (2019).

Taking the contamination injection process of Qwen2.5-7B at the label level as an example, we can see from Figure 4 that the evaluation loss of the model is only marginally improved as the training loss decreases significantly, so we consider that the injection pattern based on this setup maintains the model its own capability without causing the

[4] https://www.gdeltproject.org/

[5] https://wikipedia.org/

[6] https://scholar.google.com/

[7] https://paperswithcode.com/

[8] https://github.com/explosion/spaCy

[9] https://huggingface.co/datasets/chengxuphd/liar2

[10] https://huggingface.co/collections/Qwen/

[11] https://huggingface.co/instruction-pretrain/

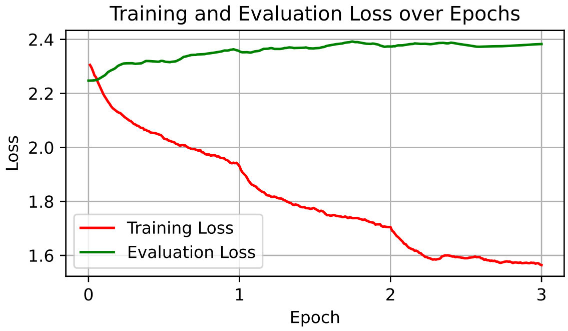

Figure 4: Training and evaluation loss of Qwen2.5-7B during the label level contamination injection, showing minimal overfitting and preserved model capability for subsequent SSA testing.

model to overfit due to the injected contamination, which is also beneficial for the simulate real pretrain scenario.

The training is conducted with PyTorch[12] and the Hugging Face Transformers[13] library, ensuring compatibility and scalability across different model configurations. For all model sizes, we adopt consistent pretraining hyperparameters unless otherwise specified: the learning rate is set to 5e-5 with a linear decay schedule, weight decay is 0.01, and we train for 3 epochs.

A.4 Resources Cost

We continue pre-trained InstrctLM and Qwen2.5 models (0.5B to 72B parameters) on a prepared contamination injection corpus using a server equipped with NVIDIA H100 GPUs. All other settings remain the same. The computational resources we spent for a single epoch pre-training are shown in Table 3.

Model	**GPU-Hours**
InstructLM-500M	1.16
InstructLM-1.3B	1.63
Qwen2.5-0.5B	0.93
Qwen2.5-1.5B	1.48
Qwen2.5-3B	2.22
Qwen2.5-7B	4.67
Qwen2.5-14B	8.67
Qwen2.5-32B	14.49
Qwen2.5-72B	25.93

Table 3: GPU cost of a single epoch pre-training (Hour).

A.5 Evaluation Details

We used the LIAR2 benchmark in this work (Xu and Kechadi, 2023, 2024), and its statistics are provided in Table 4. For the benchmark evaluations, we performed label matching by using fixed query prompts for LLMs to match answers from their responses. For example, keywords "Label:" was used to mark the starting matching position. Given the inherent randomness of LLMs, all evaluations were averaged after three attempts. Specifically, in the evaluation, we use the following prompts:

```
Classify the given political statement with six labels: 'pants-on-fire', 'false', 'mostly-false', 'half-true', 'mostly-true', 'true'
Statement: {statement}
Label:
```

Statistics	**Num.**
Training set size	18,369
Validation set size	2,297
Testing set size	2,296
Avg. statement length (tokens)	17.7
Labels	
Pants on fire	3,031
False	6,605
Barely-true	3,603
Half-true	3,709
Mostly-true	3,429
True	2,585

Table 4: The LIAR2 dataset statistics.

B Entity Shift Analysis

B.1 Implementation Details

For the entity shift procedure in our experiment, we employed gpt-4o-mini to help us with the entity shift step (OpenAI, 2024). Specifically, for our experiment dataset (the LIAR2 test set), we use the following prompt to let the model generate statements and contexts after entity shift:

```
<|system|>
Generate an entity-shifted claim and corresponding context by replacing real-world entities with fictional analogs while preserving claim semantics. Do not mention in context that the entities in it are fictional.\n
</s>
<|user|>
Return only the JSON object with the keys \"statement_revised\" and \"
```

[12] https://pytorch.org/

[13] https://github.com/huggingface/transformers

```
context\". Ensure fictional names
 are completely original and not
similar to real entities.\n Do
not mention the existence of
fictions in the generated \"
statement_revised\" and \"context
\", especially do not mention in
\"context\" that the entities are
 fictional.\n For \"
statement_revised\", claim with
real entities replaced by
fictional counterparts.\n
Identify named entities (people,
organizations, geopolitical
entities)\n"Replace each entity
with a unique fictional name\n
Maintain original grammatical
structure and claim meaning\n\n
For \"context\", brief
explanatory text for fictional
entities\n One sentence per
fictional entity, use format '[
Name] is a [description]'\n Keep
descriptions generic (e.g., 'an
American politician', 'a
pharmaceutical company')\n\n
Example 1:\n 'Statement': 'Trump
says we should protect the moat
of AI in the US.'\n '
statement_revised': 'Wannetta
says we should protect the moat
of AI in the US.'\n 'context': '
Wannetta is an American
politician.'\n Example 2:\n '
Statement': 'Pfizer suppressed
reports of vaccine side effects
.'\n 'statement_revised': 'VaxGen
 suppressed reports of vaccine
side effects.'\n 'context': '
VaxGen is a pharmaceutical
company.'\n Example 3:\n '
Statement': 'WHO releases 2025
update to the International
Classification of Diseases.'\n '
statement_revised': 'Global Human
 Health Institute (GHHI) releases
 2025 update to the International
 Classification of Diseases.'\n '
context': 'Global Human Health
Institute (GHHI) is a global
health organization.'\n\n
Statement: {statement}
</s>
```

Here is a randomly selected data entry after an entity shift procedure:

```
<|label|>
1
</s>
<|revised statement|>
Three healers from the same medical
    center 'die suddenly' in the same
     week after the medical center
    mandated a fourth Zeta-19 vaccine
     for employees.
</s>
<|context|>
Healers are medical professionals
    specializing in patient care. The
     medical center is a healthcare
    facility providing various
    medical services. Zeta-19 is a
    specific viral infection.
</s>
<|original statement|>
Three doctors from the same hospital
    'die suddenly' in the same week
    ,"" after the hospital mandated a
     fourth COVID-19 vaccine for
    employees.
</s>
```

B.2 Reliability & Cost

We employed gpt-4o-mini, one of the most cost-effective and everyday model, for the entity shift procedure in our main experiment because we wanted to test the cost and reliability of the SSA framework for daily use, and in order to ensure the validity of the data generated by the LLM, we manually reviewed all the data generated by the model one by one and manually corrected the deviation cases. Additionally, in order to explore the performance of current state-of-the-art LLMs on the entity shift procedure, we also hired o3-mini to perform the same task for comparison, and the results are provided in Table 5. We also performed the same manual review of o3-mini's output and found that o3-mini performs this task essentially perfectly, especially using reasoning_effort=high. All the reviewing was done by the authors, who are PhD-level computer science researchers in English-speaking countries.

Model Name	Cost ($)	Reliability (%)
gpt-4o-mini-2024-07-18	0.21	85.10
o3-mini-2025-01-31	6.40	99.61
- with medium	20.70	99.96
- with high	44.65	100.0

Table 5: OpenAI API cost and reliability of the entity shift procedure. Reliability refers to the percentage of entity-shifted data generated by LLM that can be directly adopted without any modification.