Exposing Hidden Intersectional Bias in LLMs

1st Badr Souani
University of Luxembourg
Luxembourg
badr.souani.ctt@outlook.com

2nd Ezekiel Soremekun
Singapore University of Technology and Design
Singapore
ezekiel_soremekun@sutd.edu.sg

3rd Mike Papadakis
University of Luxembourg
Luxembourg
michail.papadakis@uni.lu

4th Setsuko Yokoyama
Singapore University of Technology and Design
Singapore
setsuko_yokoyama@sutd.edu.sg

5th Sudipta Chattopadhyay
Singapore University of Technology and Design
Singapore
sudiptac@ieee.org

6th Yves Le Traon
University of Luxembourg
Luxembourg
yves.letraon@uni.lu

***Abstract*—*Large Language Models* (LLMs) may portray discrimination towards certain individuals, especially those characterized by multiple attributes (aka intersectional bias). Discovering *intersectional bias* in LLMs is challenging, as it involves complex inputs on multiple attributes (e.g. race and gender). To address this challenge, we propose HINTER, a testing technique that synergistically combines *mutation analysis*, *dependency parsing* and *metamorphic oracles* to automatically detect *intersectional bias* in LLMs. HINTER *generates test inputs* by systematically mutating sentences using *multiple mutations*, *validates* inputs via a *dependency invariant* and *detects biases* by checking the LLM response on the original and mutated sentences. We evaluate HINTER using six LLM architectures and 18 LLM models (GPT3.5, Llama2, BERT, etc). We found that 14.61% of the inputs generated by HINTER expose intersectional bias. Results also show that our dependency invariant reduces false positives (incorrect test inputs) by an order of magnitude. We observed that 16.62% of intersectional bias errors are *hidden*, meaning, their corresponding atomic cases do not trigger biases. Besides, we conduct a human annotation study on HINTER's discovered bias in IMDB/LLAMA2. Annotators confirm that 86.7–100% of HINTER's discovered biases are genuine discrimination. We also compare HINTER to MT-NLP for atomic bias. Results show that MT-NLP and HINTER discover atomic bias at a similar rate, but HINTER's atomic test suite is more valid than MT-NLP's test suite, according to human annotators. Overall, our work emphasize the importance of testing LLMs for intersectional bias.**

***Index Terms*—Intersectional Bias, LLM, Fairness Testing**

I. INTRODUCTION

Large Language Models (LLMs) have become vital components of critical services in society [1]. LLMs are increasingly adopted in critical domains including NLP, legal, health and programming tasks [2]. For instance, popular pre-trained models (e.g., BERT, GPT and Llama) have been deployed across multiple NLP, legal and coding tasks [3], [4]. Despite their popularity and effectiveness across various tasks, LLMs face the risk of portraying bias towards specific individuals [5]. In particular, individuals belonging to demographic groups associated with multiple protected attributes may be prone to bias [6]. Discrimination in LLMs may have serious consequences, e.g., miscarriage of justice in law enforcement [4].

To this end, we pose the following question: *How can we systematically discover intersectional (i.e., non-atomic) bias in LLMs?* An *atomic bias* occurs when a model behaves differently for two identical individuals in the same context, that only differ in one sensitive attribute (e.g., gender). Meanwhile, *intersectional biases* involve two or more different sensitive attributes (e.g., gender and race). As illustrated in Figure 1 and exemplified in Table I, we aim to *automatically generate test inputs that expose intersectional bias*. We focus on intersectional bias relating to individuals (rather than groups) since individual fairness is more fine-grained and captures the subtle unfairness relating to specific intersectional individuals. In contrast, group fairness is more coarse-grained and it often conceals individual differences within groups.

Discovering intersectional bias is challenging due to its complex nature and the huge number of possible combinations of attributes involved. Intersectional bias testing is computationally more expensive than atomic bias testing. In addition, ensuring that the generated inputs are *valid* and *indeed* expose intersectional bias is challenging. More importantly, it is vital to ensure that discovered intersectional biases are *new*, they are not already exposed during atomic bias testing. In this work, we aim to address these aforementioned challenges.

Concretely, HINTER addresses three technical challenges, namely: (1) **test generation**– systematically generate inputs that are likely to uncover intersectional bias; (2) **test validity** – automated test input validation w.r.t. original dataset; (3) **bias detection** – detect (hidden) intersectional bias at scale. To address these challenges, we propose HINTER[1], an automated black-box testing technique which synergistically combines *mutation analysis*, *dependency parsing* and *metamorphic oracles* to expose *intersectional* bias (*see* Figure 2). HINTER **generates test inputs** by systematically mutating sentences via **higher-order mutation**. Next, it **validates** inputs via a

[1] HINTER refers to (1) a tool to "***h**int*" (discover) **inter**sectional bias; and (2) it is the German word for "*behind*", implying that intersectional bias is hidden *behind* atomic bias.

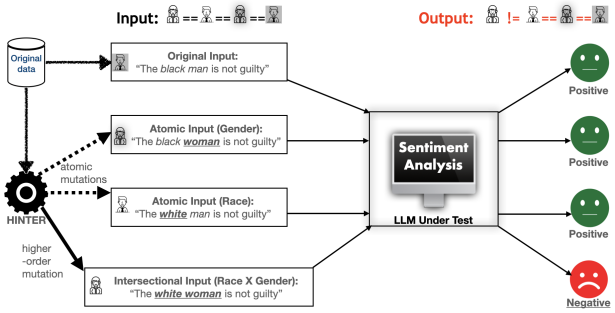


Figure 1: HINTER illustration using a Sentiment Analysis

dependency invariant which checks that the **parse trees** of the generated input and the original text are similar. Then, HINTER **detects bias** by comparing the LLM model outcome of the original text and the generated input(s). Using 18 LLMs, we demonstrate the performance of HINTER in exposing (hidden) intersectional bias. We also investigate *whether it is necessary to perform intersectional bias testing, despite atomic bias testing.* Overall, this paper makes the following contributions:

- We propose an **automated intersectional bias testing technique (HINTER)** that employs a combination of *multiple mutations, dependency parsing and metamorphic oracles*.
- **Evaluation:** We conduct an empirical study to examine the effectiveness of HINTER using three sensitive attributes, five datasets, six LLM architectures, five tasks, and 18 models.
- **Atomic vs. Intersectional Bias:** We **demonstrate the need for intersectional bias testing** by showing that 16.62% of intersectional bias errors are *hidden*, i.e., their corresponding atomic tests do not trigger biases.
- **Oracle Validation:** We validate HINTER's oracle via a human annotation study (86.7–100% oracle precision across groups, 100% mutation validity for atomic mutants).
- **MT-NLP Comparison:** We compare HINTER to MT-NLP [7] and found that HINTER's atomic test suite is more valid than MT-NLP, according to human annotators.

II. BACKGROUND

Motivating Example: Figure 1 illustrates how HINTER discovers hidden intersectional bias. Given a dataset (containing e.g., 1^{st} sentence in Figure 1), we generate higher-order mutants that expose intersectional bias in the LLM under test. In this example, the higher-order mutant (4^{th} sentence) exposes a *hidden* bias since its corresponding atomic mutations (2^{nd} and 3^{rd} sentences) do not induce any bias. Table I demonstrates an instance of *hidden intersectional bias* found by HINTER. This example shows how intersectional bias is *hidden* during atomic bias testing and demonstrates the need for intersectional bias testing. In this example the original prediction is *Negative* and the intersectional mutant's prediction changes to *Positive*. A complementary case where the original is *Positive* and the intersectional mutant yields *Negative* is provided in the supplementary material.

HINTER Overview: Figure 2 illustrates the workflow of HINTER. Given an existing dataset (e.g., training dataset), HINTER first *generates bias-prone inputs* by employing higher-order mutations. It mutates a sentence using word pairs extracted from SBIC [8], a well-known bias corpus containing 150K social media posts covering a thousand demographic groups[2]. Second, HINTER ensures that the generated inputs are *valid* by checking that it has a similar dependency parse tree with the original text (*see* Figure 3). Finally, HINTER's metamorphic oracle discovers intersectional bias by comparing the LLM outcome of the original text and the generated input. It further detects a *hidden* intersectional bias by comparing the outcomes of atomic and intersectional mutants.

Problem Formulation: Given an LLM f, HINTER aims to determine whether the inputs relating to individuals characterized by multiple sensitive attributes face bias. Thus, it generates a test suite T that includes individuals associated with multiple sensitive attributes. Consider Figure 1 that is focused on two sensitive attributes, "`race`" and "`gender`". HINTER produces an intersectional bias test suite (T_{RXG}) based on the original dataset ($Orig$) by simultaneously mutating words associated with each attribute (e.g., "*black*" to "*white*" and "*man*" to "*woman*" for "`race`" and "`gender`" attributes, respectively). We also produce an atomic bias test suite (T_R) by mutating *only* a single attribute (e.g., "white" to "black" for "race" *only*) at a time. The model outcome ($O_t = f(t)$) for every test input ($t \in T_{RXG}$) characterizes how the LLM captures an intersectional individual t for model f.

This setting allows HINTER to determine the following:

(a) *Atomic Bias* (aka individual bias) occurs when the LLM model outcomes (O) for two individuals $\{t1, t2\}$ are different ($(O_{t1} \neq O_{t2})$ even though both individual $t1$ and $t2$ are similar for the task at hand, but only differ by *exactly* one sensitive attribute (e.g., `gender`). Our definition of atomic bias require that similar individuals should be treated similarly, in line with previous literature on fairness metrics [9]–[13].

(b) Intersectional Bias occurs when the LLM model outcomes (O) for two individuals $\{t1, t2\}$ are different ($(O_{t1} \neq O_{t2})$ even though $t1$ and $t2$ are similar for the task at hand, and only differ by *at least* two sensitive attributes (e.g., `race` and `gender`). This definition is in line with the ML literature on **intersectional bias** [14], [15].

(c) Hidden Intersectional Bias refers to when atomic inputs ($atomic_r \in T_R, atomic_g \in T_G$) characterized by the test suites (T_R, T_G) are benign (trigger no bias) but their corresponding intersectional test input ($inter_{rXg} \in T_{RXG}$) from the test suites (T_{RXG}) trigger a bias. In particular, we say there is a *hidden intersectional bias* when the LLM model outcomes (O) for the atomic test inputs ($atomic_r$, $atomic_g$) are similar, but their corresponding intersectional test input ($inter_{rXg}$) produce a different outcome – $(O_{atomic_g} == O_{atomic_r}) \neq (O_{inter_{rXg}})$.

Illustrative examples for definitions (a)–(c), referencing Figure 1, are provided in the supplementary material.

[2]https://maartensap.com/social-bias-frames/

Table I: Hidden Intersectional Bias Errors found by HINTER in GPT3.5 using the IMDB/Sentiment Analysis. Mutations are marked in **bold text and underlined**. Bias (✘) is characterized by changes in GPT3.5 predictions. ("-" = Not Applicable, ✔ = No Bias/Benign, ✘ = Bias detected) A changed prediction constitutes a fairness violation. Two inputs differing only in sensitive attributes should yield the same prediction from a fair model, otherwise a difference in prediction is detected as bias.

ID	Sensitive Attributes	Mutations	Test/Mutant Type	Test Cases Exposing Hidden Intersectional Bias	GPT3.5 Prediction	Bias
19375	-	-	Original	"... There is a special heaven reserved for **people** who make the world laugh . .. **British** moviegoers will recognise the fat one from Cannon and Ball ... "	Negative	-
	Race	British→ Pakistani	**Atomic I**	"... There is a special heaven reserved for **people** who make the world laugh . .. **Pakistani** moviegoers will recognise the fat one from Cannon and Ball ... "	Negative	✔
	Gender	people→trans women	**Atomic II**	"... There is a special heaven reserved for **trans women** who make the world laugh . .. **British** moviegoers will recognise the fat one from Cannon and Ball ... "	Negative	✔
	Race & Gender	people→trans women British→ Pakistani	**Intersectional**	"... There is a special heaven reserved for **trans women** who make the world laugh ... **Pakistani** moviegoers will recognise the fat one from Cannon and Ball ...	**Positive**	✘

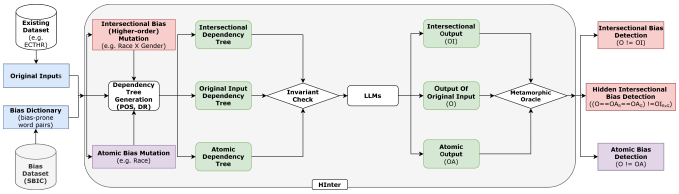


Figure 2: Workflow of HINTER showing its detection of atomic and intersectional biases in LLMs using a metamorphic oracle.

(d) Atomic vs. Intersectional Bias: We determine the differences between intersectional bias and atomic bias testing by inspecting the outcomes of our test suites ($O(T_{RXG})$ vs $O(T_R)$ vs $O(T_G)$). We inspect whether original inputs and mutations producing the atomic test suites (T_R or T_G) trigger the same biases as the intersectional bias test suites (T_{RXG}).

Definition of Terms. In this paper, we use the following terms:

- *Metamorphic-relation violation:* a test case where the model output changes between original and mutant.
- *Bias* (or *discrimination*): a metamorphic-relation violation that a human annotator judges as discriminatory.
- *Task-equivalent mutant:* a mutant whose meaning is preserved w.r.t. to the original text, according to human annotators.
- *Hidden intersectional bias:* defined in (c) above and cross-referenced in **RQ6**.
- *Benign change:* a test case where no bias is detected.

III. HINTER APPROACH

Figure 2 illustrates the workflow of our approach. Given, as *inputs*, an existing dataset, sensitive attribute(s) and bias dictionary, the *output* of HINTER is the *bias-prone test suite* which contains mutations of the original dataset and *intersectional biases* it detects. HINTER detects (hidden) intersectional bias in the following *three main steps*:

Step 1: Higher-order Mutation: HINTER transforms each original input into a bias-prone mutant via higher-order mutation. It replaces word(s) in the original input with sensitive words using the bias-prone dictionary. Our mutation operates on the full input string (spanning one or more sentences). When a term to mutate, or in the case of intersectional replacement, multiple terms appears in different sentences, replacements can occur in multiple sentences within the same text. Our bias dictionary is automatically extracted from the SBIC corpus [8] and *The bureau of Label Statistics* [16]. It contains a set of bias-prone word-pairs for each sensitive attribute, e.g., the pair ("man":"woman") mapped to the *gender* attribute.

A step-by-step example of HINTER's mutation process (race $\times$ gender) is provided in the supplementary material.

Step 2: Dependency Invariant Check: The parse tree details the structure, POS and dependency relations among words in the text (e.g., Figure 3). The main goal of this step is to check that the following **invariant** holds – *the parts of speech (POS) and dependency relations (DR) of the generated inputs are similar to the original input*. The intuition is that the generated inputs need to have a similar grammatical structure with the original input to preserve the intention and semantics of the original text. We apply invariant checking (`InvCheck`) on the both original and generated inputs (e.g., lines 19, 26 and 27 of algorithm 1).

Similarly, in atomic bias testing, the original input and modified input (mutant) are split into sentences, respectively. Next, the algorithm checks that the invariant check (`invariantCheck`) is passed. `InvCheck` first splits each text into sentences and confirms that both texts have the same number of sentences. For each sentence, it generates the parse trees for the original input and the generated test input and checks that the parse trees of both inputs are similar. It checks that each sentence in the generated input conforms to the parts of speech (POS) and dependency relations (DR) of its corresponding sentence in the original input.

This comparison is performed using `tolerantTableComp`, which tolerates small insertions or deletions (e.g., "*man*" $\rightarrow$ "*disabled man*") within a bounded divergence to validate structural consistency between original and generated sentences. Details are provided in the supplementary material in the artifact.

HINTER *discards* inputs whose parse trees are *nonconforming* to the parse tree of the original inputs, i.e., fail our dependency invariant check. However, if the parse tree of the

Algorithm 1 Intersectional Bias testing

1: {**Input:** Dataset C, LLM M, Dictionary P, Attributes $S1$, $S2$}
2: {**Output**: test suite T, intersectional bias E & hidden bias H}
3: **Function** HINTER(C, M, P, $S1$, $S2$)
4: T_{list} = [], E_{list} = [], H_{list} = []
5: {Test suite T, intersectional bias E & hidden bias H lists}
6: **for** c in C **do**
7: $O[c] \leftarrow M(c)$ {Store LLM outcome for the original text}
8: $P_1 = P[S1], P_2 = P[S2]$
9: {Get word pairs for attributes $S1, S2$ from dictionary P}
10: **for** p_1 in P_1 **do**
11: **for** p_2 in P_2 **do**
12: **if** $p_1[0], p_2[0]$ in c **then**
13: $t_1 \leftarrow c.replace(p_1[0], p_1[1])$
14: {First Atomic Mutation using Bias list P_1}
15: $t_2 \leftarrow c.replace(p_2[0], p_2[1])$
16: {Second Atomic Mutation using Bias list P_2}
17: $m \leftarrow t1.replace(p_2[0], p_2[1])$
18: {Intersectional (Higher-order) Mutation}
19: **if** $InvCheck(c, m)$ **then**
20: {Dependency invariant check}
21: $T_{list} \cup m$ {Store mutant in test suite}
22: **if** $M(m) \neq O[c]$ **then**
23: {Metamorphic Test Oracle}
24: $E_{list} \leftarrow E_{list} \cup (m, c, M(m), O[c])$
25: {Store bias inducing mutant}
26: **if** $InvCheck(c, t_1)$ **then**
27: **if** $InvCheck(c, t_2)$ **then**
28: **if** $M(t_1) \equiv O[c] \equiv M(t_2)$ **then**
29: $H_{list} \cup m$ {Store hidden bias}
30: **return** $T_{list}, E_{list}, H_{list}$

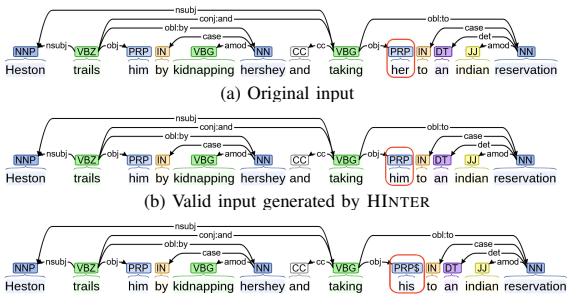
generated input (i.e., mutant) conforms with that of the original input, then HINTER adds such inputs to the resulting test suite (line 21 of algorithm 1). We refer to such inputs, that *pass* the dependency invariant check, as valid/generated inputs.

Figure 3 shows a valid mutant (b) and a discarded mutant (c). The extended walkthrough is in the supplementary material.

Step 3: Metamorphic Test Oracle: Figure 2 illustrates how HINTER detects intersectional bias. After the invariant check (**Step 2**), HINTER (algorithm 1) feeds the generated test input and its corresponding original input (line 7) into the LLM under test separately and compares the LLM outputs (line 22). HINTER detects intersectional bias (aka error), when an intersectional mutant produces a different model outcome from the original test input (line 24). Formal definitions of *benign change* and the metamorphic property are in the supplementary material. The oracle does not assume the model's output for the original input is correct. The bias is flagged whenever the outcome changes between the original, regardless of whether the original or the mutant is correct with respect to ground truth (dataset labels). Consider the original input and generated mutants in Figure 1, these inputs are similar in the context of the sentiment analysis task. Consequently, all four inputs should produce the same outcome, i.e., a "positive" label/prediction. We say that a *hidden intersectional bias* is detected (lines 22, 28-29 of algorithm 1) when there is an intersectional bias but its corresponding atomic mutants are not exposing any bias.

Figure 2 illustrates a hidden intersectional bias workflow.

A step-by-step oracle walkthrough for the example in Figure 1 is provided in the supplementary material.



(a) Original input

(b) Valid input generated by HINTER

(c) Invalid input discarded by HINTER's dependency invariant

Figure 3: Examples illustrating how HINTER's dependency invariant check works using a review from the IMDB dataset. It shows (a) an original input; (b) a valid mutation *generated* by HINTER; and (c) an invalid mutation *discarded* by HINTER. The invalid case changes the object pronoun's part-of-speech (POS) from PRP (him or her) to PRP$ (his), violating the required POS and dependency match of HINTER. POS acronyms are explained in the the supplementary material.

Table II: Details of Datasets for Training and Bias Testing

Dataset	Task Details		Training Setting			Bias Testing
	Task	Class. (#Labels)	Train	Val.	Test	(Full dataset)
ECtHR	Judgment Pred.	Multi-Label (10+1)	9,000	1,000	1,000	11,000
LEDGAR	Contract Class.	Multi-Class (100)	60,000	10,000	10,000	80,000
SCOTUS	Issue Area Pred.	Multi-Class (14)	5,000	1,400	1,400	7,800
EURLEX	Document Pred.	Multi-Label (100)	55,000	5,000	5,000	65,000
IMDB	Sentiment Analysis	Binary (2)	-	-	-	50,000

IV. EXPERIMENTAL SETUP

Research Questions (RQs): We pose the following RQs:
RQ1* HINTER*'s Effectiveness: How effective is HINTER in discovering intersectional bias? How frequently can we attribute discovered biases to the generated or original inputs?
RQ2 Oracle Validation: Do HINTER's mutants preserve semantics and discover genuine intersectional bias?
RQ3 MT-NLP Comparison: How does HINTER compare to MT-NLP, the closest prior work in metamorphic bias testing?
RQ4 Grammatical Validity: Are the inputs generated by HINTER grammatically valid?
RQ5 Dependency Invariant: What is the contribution of HINTER's dependency invariant check?
RQ6 Atomic Bias vs. Intersectional Bias: Do intersectional bias-inducing mutations and original texts *also* reveal atomic bias, and vice versa?
RQ7 Scalability: Does HINTER generalize to more than two sensitive attributes?
RQ8* HINTER*'s Sensitivity: Are HINTER's findings stable across *multiple runs*, *different prompt configurations* and varying *model confidence* thresholds?

Tasks and Datasets: Table II shows the distribution of inputs across the datasets. Table III provides details about the tasks of the datasets used in our experiments. Datasets cover formal legal text (ECtHR, LEDGAR, SCOTUS, EURLEX) and casual reviews (IMDB), spanning multi-sentence inputs and multi-class/multi-label tasks; selection rationale is in the supplementary material.

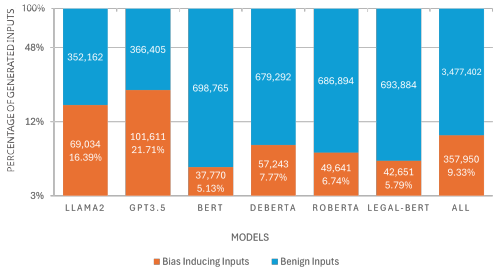


Figure 4: Atomic bias error rate (log-scaled y-axis).

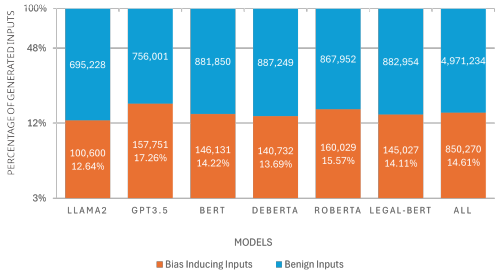


Figure 5: Intersectional bias error rate (log-scaled y-axis).

LLM Architectures and Models: Our experiments involve 18 LLM models: 2 pre-trained (`GPT-3.5-turbo-0125`, `Llama-2-7b-chat-hf`) and 16 fine-tuned across four architectures (`BERT-base`, `RoBERTa-base`, `DeBERTa`, `Legal-BERT`), covering encoder-only, decoder-only, and encoder-decoder families [3]. Further details are in our artifact.[3] Fine-tuned models achieve 64.3–88.1% micro-F1 on legal datasets; GPT-3.5 achieves 30.6–66.8% and Llama2 achieves 20.7–56.8% on legal datasets (92.9% both on IMDB); full per-model and per-dataset scores are in the supplementary material.

Sensitive Attributes: We employ three well-known sensitive attributes, namely "*race*", "*gender*" and "*body*". For atomic bias, we consider each attribute in isolation. For intersectional bias, we simultaneously combine every two attributes (i.e., $N = 2$) namely "*race X gender*", "*body X gender*" and "*body X race*". We have employed (these) three sensitive attributes due to their popularity and the prevalence of attributes in available datasets[4]. Figure 1 and Table I illustrate these attributes.

Oracle validation (RQ2): We conduct a human annotation study on IMDB to validate HINTER's oracle and mutation validity. We sample 180 cases across four groups: (1) 50 intersectional HINTER cases (25 error-inducing, 25 benign), (2) 50 atomic HINTER gender cases (25/25), (3) 50 MT-NLP gender cases (25/25), and (4) 30 HINTER-discarded cases (meaning validity only). Groups (2)–(3) are restricted to originals shared by both tools (common-originals intersection). Sampling uses a fixed seed. Annotators see only the original input, the mutated input (swapped tokens highlighted), and the task label only, with no model outputs or tool identity. Four independent non-author annotators (3 computer scientists and 1 lawyer) rate each case on two questions: *Q1* (did the swap preserve sentence meaning?) and *Q2* (should a fair model give the same answer?). In case of a 2–2 tie, a fixed adjudication rule resolves it (Q1: meaning-altered $>$ exactly equivalent $>$ unsure; Q2: may-differ $>$ same $>$ unsure). The IMDB scope is deliberate as it requires no domain expertise and it exhibits the strongest hidden intersectional bias signal in our experiments. We discuss the representativeness and limitations of this setting in the threats to validity (Section VI).

[3]https://github.com/BadrSouani/HInter

[4]Most datasets (69%, nine out of 13) have at most three attributes [6].

MT-NLP comparison (RQ3): MT-NLP [7] generates mutants by substituting sensitive-attribute terms and detects bias via a metamorphic oracle, with no validity-filtering step. The absence of input validation is a key property distinguishing MT-NLP from HINTER (section VII). Thus, we quantify that gap directly. We apply MT-NLP using its publicly available implementation on the IMDB dataset and LLAMA2 model. The experimental setting involves the IMDB originals, the *gender* attribute (female $\leftrightarrow$ male) and *atomic* mutations (MT-NLP has no intersectional mode). Comparison is restricted to the *common-originals* set, i.e., originals on which both tools produced at least one mutant. For both tools, "mutants generated" refers to the raw count before any filtering (HINTER's dependency-invariant check, MT-NLP applies no equivalent filter). Mutation quality and oracle precision for MT-NLP are reported in the human-annotation study (**RQ2**).

Multiple runs and prompts (RQ8): We execute HINTER with multiple runs and different prompts to assess its robustness and sensitivity. We focus on LLAMA2 with IMDB and SCOTUS. For RQ8.1, we repeat the pipeline ten (10) times under identical settings, configured for deterministic decoding. This dedicated stability re-run targets IMDB and LLAMA2. Three independent runs conducted under uncontrolled decoding are also reported in the supplementary material. We then run two further repetitions using two alternative prompt templates per dataset (RQ8). The labels, few-shot examples, and decoding parameters are fixed (similar to *RQ1*). Results are discussed in **RQ8** and Figure 11.

Implementation Details and Platform: HINTER and our data analysis were implemented in about 6 KLOC of Python. All experiments were conducted on a compute server with four Nvidia Tesla V100 SXM2 GPUs, two Intel Xeon Gold 6132 processors (2.6 GHz), and 768 GB of RAM. We provide the source code and experimental data for HINTER: https://github.com/BadrSouani/HInter

Additional Details: Our artifact provides a more detailed research protocol, details about our metrics and measures, prompt configurations and prompt templates with examples.

V. RESULTS

Extended analyses and additional figures for all RQs are provided in the supplementary material.

RQ1 HINTER's Effectiveness

Table III: Details of Intersectional Bias Error Rates found by HINTER. "Ⓘ" means "Intersectional Bias Error Rate" and "Ⓗ" means "Hidden Intersectional Bias Error Rate"

Datasets	Metric	Llama 2	GPT 3.5	Legal-BERT	BERT	De-BERTa	Ro-BERTa	Total
ECtHR (ECHR)	Ⓘ	51.74	59.03	5.02	4.65	5.10	3.75	11.37
	Ⓗ	(13.24)	(13.70)	(45.12)	(15.35)	(17.26)	(18.18)	(18.06)
LEDGAR (Contracts)	Ⓘ	25.35	7.04	0.00	0.00	0.00	1.05	4.58
	Ⓗ	(12.37)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(12.24)
SCOTUS (US Law)	Ⓘ	12.32	21.32	1.02	2.34	4.76	2.25	10.60
	Ⓗ	(26.33)	(19.60)	(40.87)	(13.20)	(5.86)	(8.30)	(18.92)
EURLEX (EU Law)	Ⓘ	89.62	91.00	19.99	20.28	19.02	22.97	21.66
	Ⓗ	(1.35)	(0.49)	(37.53)	(7.46)	(11.86)	(5.62)	(14.73)
IMDB (Reviews)	Ⓘ	3.52	7.12	N/A	N/A	N/A	N/A	5.43
	Ⓗ	(28.38)	(28.03)	N/A	N/A	N/A	N/A	(28.13)
All	Ⓘ	12.64	17.26	14.11	14.22	13.69	15.57	14.61
	Ⓗ	(16.99)	(18.19)	(38.46)	(8.36)	(12.45)	(6.64)	16.62

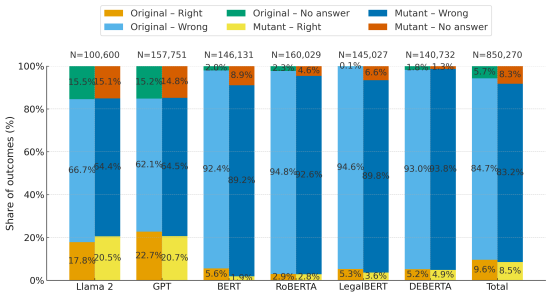


Figure 6: Accuracy of bias-inducing mutants versus bias-inducing original inputs (N = number of bias errors).

RQ1.1 Bias Error Rate: We found that *about one in seven (14.61%) inputs generated by* HINTER *revealed intersectional bias.* Table III shows that the intersectional bias error rate varies substantially by configuration, from 0.00% (LEDGAR, fine-tuned encoders) to 91.00% (EURLEX, GPT-3.5). Decoder-only models (GPT-3.5, Llama2) consistently produce higher rates than fine-tuned encoder models (BERT, Legal-BERT, RoBERTa, DeBERTa). On EURLEX, GPT-3.5 reaches 91.00% and Llama2 reaches 89.62%, while fine-tuned encoders remain in the 19–23% range. We attribute this gap to task-specific fine-tuning, which anchors encoder predictions to dataset-level label patterns and reduces sensitivity to surface-level token substitutions. On LEDGAR, fine-tuned encoders show 0.00%, consistent with low lexical overlap between bias-attribute terms and contract classification. 39.88% of distinct original inputs trigger at least one intersectional bias; per-model aggregate breakdown is in the supplementary material. Figure 4 illustrates the atomic bias error rate across the models. GPT-3.5 shows the highest percentage of bias-inducing outputs at 21.71%, significantly above the overall model average of 9.33%. We also inspected the proportion of distinct original inputs that resulted in generating at least one mutant with atomic bias errors. GPT-3.5 again presents the highest susceptibility, with 35.78% of original inputs producing bias-inducing mutants. Overall, approximately 38.43% (more than one-third) of original inputs led to errors. Overall, results show that HINTER is effective in exposing (hidden) intersectional bias across tested models.

HINTER *is effective in exposing intersectional bias: 14.61% of the inputs generated by* HINTER *exposed intersectional bias, and 16.62% of the intersectional bias errors exposed by* HINTER *are hidden.*[a]

[a] 14.61% refers to the raw metamorphic-relation violation rate; human-validated oracle results and discrimination rate are reported in **RQ2**.

An illustrative example of hidden intersectional bias found by HINTER in LLAMA 2 on the EURLEX dataset is provided in the supplementary material.

RQ1.2 Direction of Bias: We observed that *intersectional bias-inducing original inputs are equally likely to be correct or incorrect as the bias-inducing mutants, i.e., the inputs generated by* HINTER. Figure 6 shows that across all datasets and models, 84.7% of intersectional bias-inducing original inputs and 83.2% of the intersectional bias-inducing inputs generated by HINTER (mutants) lead to *incorrect* LLM outcomes. The proportion of the correct inputs for the bias-inducing original and mutants is also close: 9.6% of original inputs inducing intersectional bias produce correct LLM outcomes. In contrast, 8.5% of the bias-inducing mutants produce correct LLM outcomes. This result also holds for *hidden intersectional bias*, where we observed that about four in five bias-inducing original inputs (79.3%) or bias-inducing mutants (79.4%) lead to *incorrect* model outcomes. These results suggest that *intersectional bias-inducing original inputs and* HINTER*'s bias-inducing mutants are similarly likely to lead to correct or incorrect model outcomes*.

However, we observed that for inputs leading to hidden intersectional bias, the original inputs are about 51% more likely to be valid ("right") than the mutants (9.20% vs. 13.92%). We also found that bias-inducing mutants are more likely to elicit an empty LLM response ("no answer") than the original inputs (*see* Figure 6). For intersectional bias-inducing mutants, the mutants are 45% more likely to result in an empty LLM response than the original inputs (8.3% vs. 5.7%). This observation also holds for hidden intersectional bias. We found that the LLM is 69% (11.37% vs. 6.74%) more likely not to respond ("no answer") for mutants than for the original inputs. These findings suggest that HINTER induced more inaccuracies and empty responses in LLMs than the original input during (hidden) intersectional bias testing. In summary, these findings demonstrate HINTER's ability to generate inputs that induce biased LLM behaviours beyond the original dataset.

HINTER *induces errors beyond the original dataset:* HINTER*'s bias-inducing mutants are 51% more likely to cause incorrect LLM outcomes and 69% more likely to lead to an empty LLM response than original inputs.*

RQ2 Oracle Validation:

RQ2.1 Oracle Precision: We found that *the vast majority of* HINTER*'s metamorphic violations represent genuine bias.*

Table IV: Human annotation study results (IMDB). Validity = Q1 "exactly equivalent"; Precision = Q2 "same" given error-inducing AND Q1=equivalent. 95% CI: Wilson score interval. n_{prec}: cases used for precision. Fleiss κ: Q1=0.386, Q2=0.519.

Group	n	Validity	Val. 95% CI	n_{prec}	Precision	Prec. 95% CI
Intersectional HINTER	50	66.0%	[52.2%, 77.6%]	15	86.7%	[62.1%, 96.3%]
Atomic HINTER (gen.)	50	100.0%	[92.9%, 100.0%]	25	100.0%	[86.7%, 100.0%]
MT-NLP (gender)	50	84.0%	[71.5%, 91.7%]	21	100.0%	[84.5%, 100.0%]
HINTER-discarded	30	46.7%	[30.2%, 63.9%]	N/A	N/A	N/A

For atomic gender mutants, HINTER's precision is 100.0% [86.7%, 100.0%] (n=25), confirming that every semantically valid flag reflects a true fairness violation. For intersectional HINTER, precision is 86.7% [62.1%, 96.3%] (n=15). 13 of the 15 intersectional cases confirmed as both error-inducing and semantic-preserving by annotators were judged as genuine discrimination by annotators. For MT-NLP, precision is also 100.0% [84.5%, 100.0%] (n=21) using a subset of 50 metamorphic violations reported in **RQ3** (Table V). These results validate HINTER's metamorphic oracle, its violations are mostly genuine fairness violations.

RQ2.2 Mutation Validity: We found that HINTER*'s dependency invariant enforces meaningful semantic filtering beyond grammatical correctness*. Annotated valid intersectional mutants reach 66.0% validity [52.2%, 77.6%] and valid atomic mutants 100.0% [92.9%, 100.0%], Meanwhile HINTER's discarded mutants achieve only 46.7% annotator validity [30.2%, 63.9%]. The gap between accepted (66–100%) and discarded (46.7%) inputs shows that the invariant check eliminates mutations that shift sentence meaning, not merely grammatically incorrect ones. The lower validity for intersectional mutants (66.0%) compared to atomic ones (100.0%) is expected since simultaneously substituting multiple attributes increases the likelihood of altering semantics.

HINTER*'s metamorphic violations are mostly valid bias instances. Annotators mark 100% of* HINTER*'s atomic violations as genuine bias and 86.7% [62.1%, 96.3%] (n=15) of its intersectional violations as discriminatory.*

RQ3 MT-NLP Comparison: We found that HINTER *substantially reduces the resulting test-suite in comparison to* MT-NLP*, while maintaining a comparable violation rate*. Table V shows that on (26,943) common originals, MT-NLP generates 79,272 mutants and discovers 165 errors, while HINTER generates 27,877 mutants after the dependency invariant check and detects 74 errors (0.27%). The quality difference is confirmed by the human-annotation study (**RQ2**). MT-NLP's mutation validity is 84.0%, versus 100.0% for HINTER's atomic mutants, and oracle precision is 100.0% for both tools (Table IV).

HINTER *generates a test suite which is 2.8× fewer mutants than* MT-NLP*, but with comparable bias rate.*

Table V: HINTER vs. MT-NLP using atomic gender mutations (female↔male), IMDB and LLAMA2. Bias error rate = detected errors / retained mutants. For MT-NLP, no validity filter is applied: valid mutants = generated mutants.

Tool	Generated (pre-filter)	Retained	Flagged	Flag Rate
MT-NLP	79,272	79,272	165	0.21%
HINTER	167,176	27,877	74	0.27%

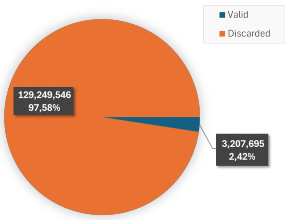


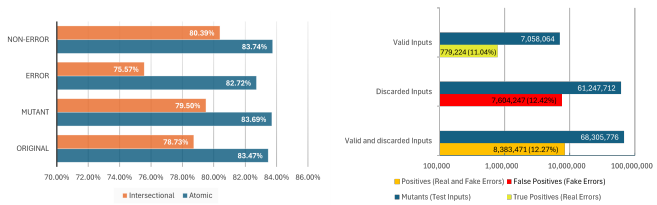
Figure 7: The distribution of valid mutants (inputs that passed invariant check) and discarded mutants (inputs that failed it).

RQ4 Grammatical Validity: We observed that *the inputs generated by* HINTER *are as valid as the original human-written text*. Figure 8 (a) shows that *the inputs generated by* HINTER *have similar grammatical scores as the original (human-written) text* for both atomic and intersectional bias testing campaigns. For intersectional bias testing, the weighted mean score of the original text (78.73%) is similar to that of the inputs generated by HINTER (79.50%). We also found that *the grammatical validity of error-inducing inputs is slightly lower than that of benign, non-error-inducing inputs* for both the original inputs and HINTER's generated input (e.g., 82.72 vs 83.74 for atomic bias testing). The Grammarly scores for generated inputs (mutants) remain similar to their corresponding original inputs (*see* Figure 8 (a)). These results show that HINTER generates grammatically valid inputs and preserves the grammatical validity of the original input.

HINTER *preserves the grammatical validity of the original (human-written) inputs:* HINTER*'s generated inputs are as grammatically valid as the original (human-written) texts.*

RQ5 Dependency Invariant

RQ5.1 Number of Valid vs. Discarded Inputs: We found that HINTER*'s invariant check is effective in identifying invalid mutated inputs:* HINTER *discards the majority (97.58%) of mutated inputs as invalid because their dependency parse tree does not align with the parse tree of the reference original text.* Figure 7 shows that most input mutations (129M/132M) result in a different dependency parse tree in comparison to the original text. Intuitively, this means that the majority of generic mutations create inputs that have a different meaning or intent in comparison to the original reference text. We observed that *only* 2.42% of input mutations *pass* our invariant check. This suggests that generic mutations (without our invariant



(a) Grammatical Validity (b) True and false positives

Figure 8: Details of (a) the proportion of HINTER's generated inputs that are grammatically valid vs. the original (human-written), (b) the true and false positives generated by HINTER, i.e., with/without dependency invariant.

check) mostly result in invalid test inputs.[5] Intersectional bias mutations and longer texts are more likely to be invalid (fail our dependency invariant) due to their complexity. These results emphasize the importance of HINTER's invariant check.

> HINTER*'s invariant check effectively identifies invalid inputs: It discarded millions (129M/132M) of mutants whose dependency parse trees do not conform to the original text.*

RQ5.2 False Positives: Without HINTER*'s dependency invariant, developers would need to inspect about 10 times (10X) as many fake errors as real errors*. HINTER saves developers a huge amount of time and effort, it reduces the biases testing effort by up to 10 times (779,224 vs. 7,604,247). Generic mutations, without dependency invariant, produce millions of false positives – about nine out of every ten bias errors are false positives: Figure 8 (b) shows that 90.71% (7,604,247/8,383,471) errors produced by generic mutations are fake. These errors are produced by invalid mutants that do not conform with the original input. We also observed that intersectional bias testing produces more false positives than atomic bias testing (6,865,033 vs. 739,214) due to the increasing complexity of intersectional mutants. This demonstrates the contribution of HINTER's dependency invariant.

> HINTER*'s dependency invariant saves developers from inspecting millions of false positives (7,604,247) which are 10 times (10X) as many as true positives (779,224).*

RQ5.3 Validity of Discarded inputs: This experiment employs Grammarly [17] and a similar setting as **RQ4**. We sample an additional 67K discarded mutants to compare to the 68K valid inputs sampled in **RQ4** for valid/generated inputs. Similar to **RQ4**, we randomly sample about 10K original inputs and their corresponding 67K discarded mutants, while aiming for balance across sensitive attribute.

[5]Generic mutation-based bias testing approaches (e.g., MT-NLP [7]) mutate inputs without a validity check.

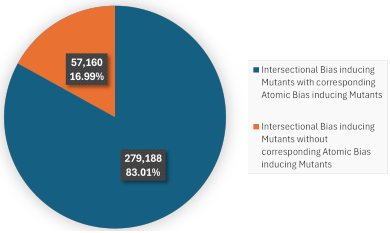


Figure 9: The distribution of intersectional and atomic bias-inducing mutants.

We found that *valid inputs are up to 8.31% more grammatically valid than the inputs discarded by* HINTER*'s invariant check.* The valid inputs generated by HINTER (i.e., pass its invariant check) maintain similar grammatical validity as the original input (79.50% vs. 78.73%). In contrast, discarded mutants (that fail invariant check) do not preserve the grammatical validity of the corresponding original texts. *Discarded mutants have (up to 6.4%) lower grammatical correctness scores than the original texts* (78.44% vs. 73.40%). These results demonstrate that HINTER's dependency invariant preserves the validity of the original inputs.

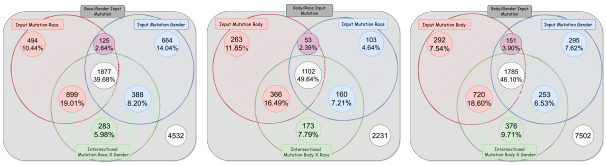
> HINTER*'s invariant check preserves the validity of the original text: Generated inputs are as valid as original inputs but up to 8.31% more grammatically valid than discarded inputs.*

RQ6 Atomic Bias vs. Intersectional Bias:

RQ6.1 Bias-inducing Mutants: In absolute terms, *atomic bias testing misses about five times (5X) as many bias-inducing mutants as intersectional bias testing (57,160 vs. 11,705).* Figure 9 shows that 16.99% (57,160/336,348) of intersectional bias-inducing mutants do not have any corresponding atomic bias. This implies that *atomic bias testing is insufficient to reveal intersectional bias instances.* Meanwhile, we found that 83.43% (58,925/70,630) of atomic bias-inducing mutants have a corresponding intersectional bias-inducing mutants. This suggests that intersectional bias testing exposes about four in five atomic bias instances. These results emphasize that intersectional bias testing complements atomic bias testing.

> *Atomic bias testing misses five times (5X) as many bias-inducing mutants as intersectional bias testing (57,160 vs. 11,705).*

RQ6.2 Bias-inducing Original Inputs: Up to one in ten (9.71%) bias-inducing original inputs strictly lead to intersectional bias. Across all settings, about 5.98% to 9.71% of bias-inducing original inputs strictly trigger *only* intersectional bias. As an example, Figure 10(c) shows that 9.71% (376) of bias-inducing original inputs strictly lead to intersectional bias (Body X Gender) sensitive attribute. We observed that *up to half of all*



(a) Race X Gender (b) Body X Race (c) Body X Gender

Figure 10: Venn diagram showing the distribution of bias-inducing original inputs leading to atomic vs. intersectional errors.

bias-inducing original inputs lead to both intersectional bias and atomic bias. Notably, 49.64% of bias-inducing original inputs lead to both intersectional bias as shown in Figure 10(b) (Body X Race), as well as atomic bias (Body only and Race only). Finally, about 4.64% to 14.04% of bias-inducing original inputs strictly lead to atomic bias. This suggests that some intersectional bias-inducing original inputs may not induce atomic bias, and vice versa.

Up to one in ten (9.71%) original inputs are strictly intersectional bias-inducing, their mutations do not trigger atomic bias.

Table VI: Scalability results for HINTER on up to three sensitive attributes, using the first 10K IMDB reviews and LlaMA2. (# means number of bias errors)

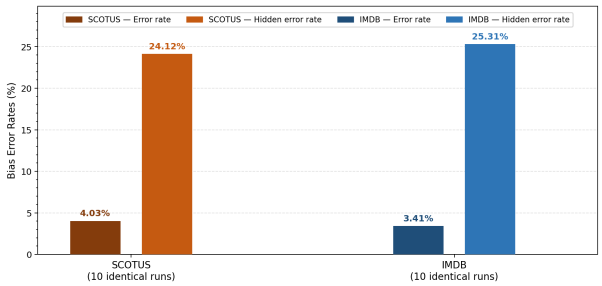
Bias Setting	#Sens. Attr.	Gen. Inputs	Invariant Check # (Passed Rate)	Inter. Bias # (Error Rate)	# Hidden Inter. Bias # Error Rate)
Atomic	1	500,157	62,601 (12.52%)	1,536 (2.45%)	/
2-Inters.	2	9,447,982	132,150 (1.40%)	3,899 (2.95%)	868 (22.26%)
3-Inters.	3	46,507,573	249,886 (0.54%)	9,851 (3.94%)	331 (3.36%)

RQ7 Scalability: Results show that HINTER *is effective in discovering intersectional bias with three simultaneous sensitive attributes (N=3)*. Scaling from N=2 to N=3 increases mutant generation from 9,447,982 to 46,507,573 candidates (a 4.9× increase, Table VI), which is why we restrict the N=3 experiment to the first 10K inputs on IMDB/LLAMA2. Notably, Table VI shows that HINTER discovers intersectional bias with three simultaneous sensitive attributes (aka 3-intersectional) at a slightly higher rate than intersectional bias with three simultaneous sensitive attributes (aka 2-intersectional) – 3.95% vs. 2.95%. This result suggests that HINTER scales for more than two sensitive attributes. However, we observed that the rate of hidden bias errors is significantly (6X) lower for three simultaneous sensitive attributes (3-intersectional). We attribute this to the reduced likelihood that each of the three atomic inputs for each sensitive attribute does *not* induce bias, but the intersection of all three induces a bias. Table VII shows a hidden intersectional bias with three simultaneous sensitive attributes found by HINTER in LlaMA2. In summary, these results imply that HINTER scales to more than two sensitive attributes, but it is more difficult to reveal hidden intersectional bias as the number of sensitive attributes increases.

Table VII: An Example of Hidden Intersectional Bias involving three sensitive attributes found by HINTER in **Llama 2** and the IMDB (#12) dataset for Sentiment Analysis. Mutations are marked in **bold text and underlined**. Bias (✘) is characterized by changes in Llama 2 predictions. ("-" = Not Applicable, ✔ = No Bias/Benign, ✘ = Bias detected)

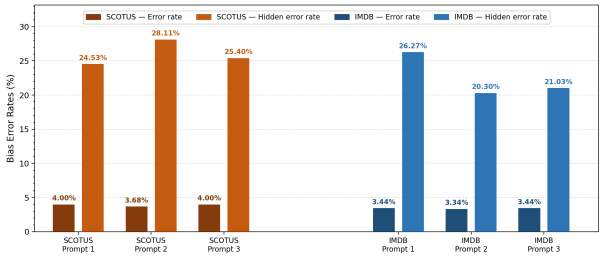
Input (Attr.)	Mutations	Test Cases Exposing Hidden Intersectional Bias	Llama 2 Pred. (Bias)
Original	-	"... **People** who have enjoyed ... Carver is **German** ...**he**'s not really staying ..."	Negative (-)
Atomic I (Body)	people→people with mental disorder	"... **People with mental disorder** who have enjoyed ... Carver is **German** ...**he**'s not really staying ..."	Negative (✔)
Atomic II (Race)	German→Syrian	"... **People** who have enjoyed ...Carver is **Syrian** ... **he**'s not really staying ..."	Negative (✔)
Atomic III (Gender)	He→She	"... **People** who have enjoyed ... Carver is **German** ...**She** is not really staying ..."	Negative (✔)
Inters. (Body, Race & Gender)	people→people with mental disorder, German →Syrian, He→She	"... **People with mental disorder** who have enjoyed ... Carver is **Syrian** ...**She** is not really staying ..."	**Positive** (✘)

	SCOTUS Error rate	SCOTUS Hidden error rate	IMDB Error rate	IMDB Hidden error rate
Average	4.03%	24.12%	3.41%	25.31%
Median	4.03%	24.12%	3.41%	25.31%
Variance	0.00%	0.00%	0.00%	0.00%
S.D.	0.00%	0.00%	0.00%	0.00%



(a) Multiple runs (10)

	SCOTUS Error rate	SCOTUS Hidden error rate	IMDB Error rate	IMDB Hidden error rate
Average	3.89%	26.02%	3.41%	22.53%
Median	4.00%	25.40%	3.44%	21.03%
Variance	0.03%	3.48%	0.00%	10.59%
S.D.	0.18%	1.87%	0.06%	3.25%



(b) Multiple prompts

Figure 11: Stability of HINTER's bias error rates across (a) ten deterministic runs and (b) different prompt configurations (LLAMA2, IMDB and SCOTUS).

HINTER *is effective in discovering intersectional bias involving three sensitive attributes. However, discovering hidden intersectional bias is challenging as the number of sensitive attributes increases.*

RQ8 HINTER's Sensitivity

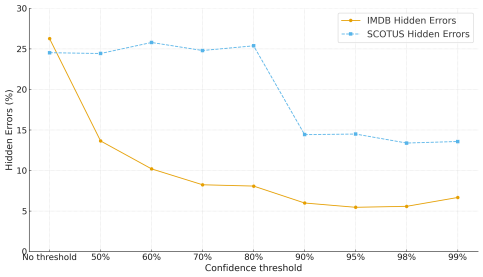
RQ8.1 Stability across Multiple Runs: We found that HINTER*'s bias error rates are stable across all ten runs*. Because decoding is fully deterministic, all ten executions produce bit-for-bit identical outputs, with the only difference being inference time. Consequently, the variance in HINTER's error rates is zero (0) across every metric and both datasets (*see* Figure 11(a)). Per-instance hidden-label agreement is 100%. Every case detected as hidden intersectional bias in the first run is detected identically in all nine subsequent runs, with no disagreement observed. This result confirms that HINTER's findings are not an artifact of stochastic decoding.

> HINTER*'s effectiveness is stable across ten deterministic runs, with zero variance in error rates for both intersectional bias and hidden intersectional bias, and 100% per-instance hidden-label agreement across all runs.*

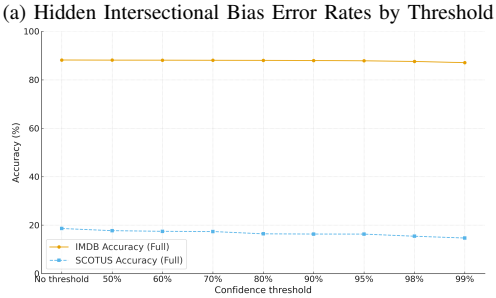
RQ8.2 Sensitivity to Different Prompts: The two alternative prompt templates vary the phrasing of the task instruction (e.g., explicit enumeration of label names versus an implicit "classify as" directive) while keeping the label set, few-shot examples, and decoding configuration identical to **RQ1**. Full prompt texts are provided in the artifact. We found that HINTER*'s effectiveness is mostly stable across multiple prompts, but varies more for hidden intersectional bias than intersectional bias.* HINTER's effectiveness is particularly stable for intersectional bias error. For instance, HINTER's error rate is between 3.34% and 3.44% for IMDB, with a mean error rate of 3.41 and zero (0) variance. However, we observe a higher variance for hidden intersectional bias than intersectional bias, e.g., zero (0) versus 10.59 for IMDB. Additionally, the stability of hidden intersectional bias error rate of HINTER may be influenced by the dataset: For example, IMDB exhibits a higher variance than SCOTUS for hidden intersectional bias (3.48 vs. 10.59). This result implies that HINTER's performance is mostly stable across multiple prompts.

> HINTER*'s effectiveness is stable across multiple prompts, but varies more for hidden intersectional bias than intersectional bias.*

RQ8.3 Impact of Model Confidence on Bias: Results show that *the proportion of* HINTER*'s detected (hidden) intersectional bias decreases as the model confidence threshold increases.* This result holds for intersectional bias in the IMDB dataset and hidden intersectional bias (*see* Figure 12 (a)) in both datasets. However, for SCOTUS, the proportion of intersectional bias increases as the model confidence increases, especially above the 90% confidence threshold. This implies that the LLM was confident in portraying intersectional bias for SCOTUS. We attribute this behaviour to the complex nature of SCOTUS (a multi-class classification dataset) and the low model accuracy of LlaMA2 for SCOTUS (*see* Figure 12 (b)). Overall, as the LLM's confidence increases, the proportion of hidden intersectional bias decreases. This suggests that model confidence can mitigate the effect of (hidden) intersectional bias.



(a) Hidden Intersectional Bias Error Rates by Threshold



(b) Accuracy of all Inputs by Threshold

Figure 12: Impact of confidence threshold on errors (Llama 2) showing IMDB (orange) and SCOTUS (blue). The x-axis is the confidence threshold (leftmost "No threshold" = 0%)

> *The proportion of hidden intersectional bias detected by* HINTER *decreases as the LLM confidence increases.*

VI. Limitations and Threats to Validity

We discuss the limitations of this work and the threats to the validity of HINTER and its findings.

Construct Validity: This includes automated bias detection in LLM responses and the metrics employed. We employ a metamorphic test oracle with system prompts and few-shot prompting to ensure LLM responses are within allowed outcomes, and standard measures (error rate, proportion of error-inducing inputs). These oracle settings and measures are commonly employed in bias testing literature [7], [18]–[20].

Internal Validity: The threat to internal validity refers to whether our implementation of HINTER actually discovers intersectional bias. To mitigate this threat, we have tested our implementation, conducted code review and validated HINTER's detected outputs via the structured human annotation study (**RQ2**; n=180, 4 annotators, Fleiss κ=0.386–0.519), and examined grammatical validity of generated inputs (**RQ4**, **RQ5**). Our dependency invariant check automatically assesses grammatical and structural conformance (matching POS tags and dependency relations). Its contribution is assessed in **RQ5**. HINTER's dependency invariant enforces structural conformance but not full semantic equivalence. Oracle precision and mutation validity are empirically supported by **RQ2**

(IMDB only, extension to other datasets is future work). The MT-NLP comparison (**RQ3**) is scoped to the *gender* attribute and *atomic* level on IMDB (MT-NLP has no intersectional mode), restricted to common-originals (N=26,943). MT-NLP was designed for shorter inputs and has more substitution opportunities on multi-sentence IMDB reviews. Key internal-validity threats:

- *Mutation design:* word pairs from SBIC [8]; dependency invariant filters structure-altering candidates. *Mitigated* (**RQ5**).
- *Oracle sensitivity:* metamorphic oracle may detect benign surface changes. *Mitigated*: 86.7–100% precision (**RQ2**).
- *Syntactic-validation limits:* invariant enforces POS-tag and dependency conformance, not content-level semantics. *Partially mitigated*: 66–100% validity for accepted vs. 46.7% for discarded (**RQ2**).
- *Stochasticity: Mitigated*: 100% label agreement across ten deterministic runs (**RQ8.1**).
- *Distribution shift:* SBIC terms may not match legal datasets; assessed indirectly via **RQ4** and **RQ5**. In legal texts, sensitive-term substitutions (e.g., nationality or disability status in ECtHR/SCOTUS cases) may carry task-relevant meaning beyond surface form; this remains an unmitigated limitation.

External Validity: The main threat to external validity is the generalizability of HINTER to other LLMs, datasets and tasks. We mitigate this by employing models covering the three main LLM architectures: encoder-only (BERT, Legal-BERT, RoBERTa), decoder-only (GPT-3.5, Llama2), and encoder-decoder (DeBERTa), across five datasets with well-studied sensitive attributes [6] and complex tasks [3]. HINTER can be easily adapted to new LLMs, datasets and tasks. A remaining limitation is that our experiments use few-shot prompting, which may not generalise to other prompting techniques. The human annotation study (**RQ2**) and MT-NLP comparison (**RQ3**) are both scoped to IMDB/LLAMA2. Four annotators participated, none of whom are authors (Fleiss κ=0.386–0.519). A remaining threat is that annotators reviewed highlighted excerpts rather than full documents. Local-context presentation may affect validity judgments for long inputs.

Dictionary Completeness: Our bias dictionary may not cover all bias-prone word pairs. We mitigate this by using SBIC [8], a well-known corpus of 150K social media posts. Extending the dictionary to new sensitive attributes is achievable within few LoC.

Higher-order mutations: Intersectional bias can arise from mutations across more than two sensitive attributes. While HINTER focuses on N=2, **RQ7**shows it scales to N=3 (race, gender, body) on IMDB/LLAMA2 (first 10K inputs for tractability). Running higher orders at scale is computationally intensive.

VII. Related Work

Intersectional Bias Testing: Previous research has shown that discrimination is multifaceted in the real world [6], [14], [21]–[25]. Recent work analysed intersectional stereotypes in LLMs [26], biased narratives [27], intersectional bias in word representations [28] and NLP models [29], and hiring disparities [30], consistently finding debiasing insufficient for compounding biases. Fairness frameworks for multiple protected attributes [31], [32] target structured data during training and do not reveal hidden intersectional bias in LLMs.

Metamorphic Testing in NLP: NLPLego [33], [34] applies metamorphic relations to discover functional errors, not bias. MT-NLP [7] is the closest approach but is limited to atomic bias testing and employs generic mutations without input validation, leading to numerous false positives (Section 5, **RQ5**). We directly quantify this gap in **RQ3** (Table V): on the same IMDB inputs, HINTER's dependency invariant reduces the test-input pool by 2.8× while maintaining a comparable bias error rate (0.27% vs. 0.21%), and our human annotation study (**RQ2**) shows MT-NLP's mutation validity is 84.0% versus 100% for HINTER's atomic mutants.

NLP Test Validity: NLP testing approaches validate test cases using templates (e.g., Checklist [18]), grammars (e.g., ASTRAEA [19]), or ML techniques [35]–[37]. However, these methods require manual curation and are not directly applicable to new or unseen inputs.

Dependency Parsing: HINTER's dependency invariant is similar to structural invariants in machine translation testing [38]–[40], which use dependency parsing to validate test inputs; however, these approaches are designed for MT systems and do not apply to bias testing.

Bias in LLMs: Previous work highlights challenges in evaluating bias in LLMs [41]–[43]. These works rely on static datasets rather than automated test generation In contrast, HINTER automatically generates test cases to uncover hidden intersectional biases, making bias evaluation more scalable.

VIII. Conclusion

This paper presents HINTER, an automated bias testing technique that expose intersectional bias in LLMs. It generates valid test inputs that uncover *hidden* intersectional bias via a combination of higher-order mutation analysis, dependency parsing and metamorphic oracle. We evaluate the effectiveness of HINTER using five datasets and 18 LLMs and demonstrate that 14.61% of inputs generated by HINTER expose intersectional bias. We also show that intersectional bias testing is unique and complements atomic bias testing: 16.62% of intersectional bias found by HINTER are hidden. Furthermore, HINTER's dependency invariant reduces the number of bias instances developers need to inspect by an order of magnitude (10X). We hope that this work will enable LLM practitioners to discover intersectional bias. Finally, we provide our experimental data, supplementary materials and HINTER's implementation to support replication and reuse in our repository: https://github.com/BadrSouani/HInter

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[2] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar *et al.*, "Holistic evaluation of language models," *Transactions on Machine Learning Research*, 2022.

[3] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. Katz, and N. Aletras, "LexGLUE: A benchmark dataset for legal language understanding in English," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4310–4330. [Online]. Available: https://aclanthology.org/2022.acl-long.297

[4] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias: There's software used across the country to predict future criminals and it's biased against blacks," *URL https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing*, 2019.

[5] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[6] U. Gohar and L. Cheng, "A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges," 2023.

[7] P. Ma, S. Wang, and J. Liu, "Metamorphic testing and certified mitigation of fairness violations in nlp models." in *IJCAI*, vol. 20, 2020, pp. 458–465.

[8] M. Sap, S. Gabriel, L. Qin, D. Jurafsky, N. A. Smith, and Y. Choi, "Social bias frames: Reasoning about social and power implications of language," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5477–5490. [Online]. Available: https://aclanthology.org/2020.acl-main.486

[9] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[10] B. Friedman and H. Nissenbaum, "Bias in computer systems," *ACM Transactions on information systems (TOIS)*, vol. 14, no. 3, pp. 330–347, 1996.

[11] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the international workshop on software fairness*, 2018, pp. 1–7.

[12] K. Crawford, "The trouble with bias, in 31th conference on neural information processing systems (nips)," *Long Beach, CA, USA*, 2017.

[13] A. Narayanan, "Translation tutorial: 21 fairness definitions and their politics," in *Proc. Conf. Fairness Accountability Transp., New York, USA*, vol. 1170, 2018, p. 3.

[14] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 77–91.

[15] C. D'Ignazio and L. F. Klein, *Data feminism*. MIT press, 2020.

[16] U.S. Bureau of Labor Statistics, "Current population survey, table a-11. employment, hours, and earnings by industry," https://www.bls.gov/cps/cpsaat11.htm, 2025, accessed July 21, 2025.

[17] B. Hoover, D. Lider, A. Shevchenko, and M. Lytvyn, "Grammarly: Free writing AI Assistance," https://www.grammarly.com/, 2009, accessed: 2023-06-19.

[18] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of nlp models with checklist," in *Annual Meeting of the Association for Computational Linguistics*, 2020.

[19] E. Soremekun, S. Udeshi, and S. Chattopadhyay, "Astraea: Grammar-based fairness testing," *IEEE Transactions on Software Engineering*, vol. 48, no. 12, pp. 5188–5211, 2022.

[20] S. S. Rajan, E. Soremekun, Y. Le Traon, and S. Chattopadhyay, "Distribution-aware fairness test generation," *Journal of Systems and Software*, vol. 215, p. 112090, 2024.

[21] E. Soremekun, M. Papadakis, M. Cordy, and Y. L. Traon, "Software fairness: An analysis and survey," *arXiv preprint arXiv:2205.08809*, 2022.

[22] K. Yang, J. R. Loftus, and J. Stoyanovich, "Causal intersectionality for fair ranking," *arXiv preprint arXiv:2006.08688*, 2020.

[23] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau, "Fairvis: Visual analytics for discovering intersectional bias in machine learning," in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2019, pp. 46–56.

[24] K. Crenshaw, "Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics," *University of Chicago Legal Forum*, p. 139, 1989.

[25] P. H. Collins, *Intersectionality as critical social theory*. Duke University Press, 2019. [Online]. Available: http://www.jstor.org/stable/j.ctv11hpkdj

[26] W. Ma, B. Chiang, T. Wu, L. Wang, and S. Vosoughi, "Intersectional stereotypes in large language models: Dataset and analysis," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 8589–8597. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.575/

[27] H. Devinney, J. Björklund, and H. Björklund, "We don't talk about that: Case studies on intersectional analysis of social bias in large language models," in *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, A. Faleńska, C. Basta, M. Costa-jussà, S. Goldfarb-Tarrant, and D. Nozza, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 33–44. [Online]. Available: https://aclanthology.org/2024.gebnlp-1.3/

[28] Y. C. Tan and L. E. Celis, "Assessing social and intersectional biases in contextualized word representations," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/201d546992726352471cfea6b0df0a48-Paper.pdf

[29] J. Lalor, Y. Yang, K. Smith, N. Forsgren, and A. Abbasi, "Benchmarking intersectional biases in NLP," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 3598–3609. [Online]. Available: https://aclanthology.org/2022.naacl-main.263/

[30] K. Wilson and A. Caliskan, "Gender, race, and intersectional bias in resume screening via language model retrieval," 2024. [Online]. Available: https://arxiv.org/abs/2407.20371

[31] H. Tian, B. Liu, T. Zhu, W. Zhou, and P. S. Yu, "Multifair: Model fairness with multiple sensitive attributes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 3, pp. 5654–5667, 2025.

[32] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, "Fairness improvement with multiple protected attributes: How far are we?" 2024. [Online]. Available: https://arxiv.org/abs/2308.01923

[33] P. Ji, Y. Feng, W. Huang, J. Liu, and Z. Zhao, "Nlplego: Assembling test generation for natural language processing applications." *CoRR*, 2023.

[34] ——, "Intergenerational test generation for natural language processing applications," *arXiv preprint arXiv:2302.10499*, 2023.

[35] A. Liu, S. Swayamdipta, N. A. Smith, and Y. Choi, "Wanli: Worker and ai collaboration for natural language inference dataset creation," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 6826–6847.

[36] P. West, C. Bhagavatula, J. Hessel, J. Hwang, L. Jiang, R. Le Bras, X. Lu, S. Welleck, and Y. Choi, "Symbolic knowledge distillation: from general language models to commonsense models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 4602–4625.

[37] G. Yang, M. Haque, Q. Song, W. Yang, and X. Liu, "Testaug: A framework for augmenting capability-based nlp tests," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3480–3495.

[38] P. He, C. Meister, and Z. Su, "Structure-invariant testing for machine translation," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 961–973.

[39] Q. Zhang, J. Zhai, C. Fang, J. Liu, W. Sun, H. Hu, and Q. Wang, "Machine translation testing via syntactic tree pruning," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 5, pp. 1–39, 2024.

[40] Z. Sun, J. M. Zhang, M. Harman, M. Papadakis, and L. Zhang, "Automatic testing and improvement of machine translation," in *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, 2020, pp. 974–985.

[41] I. Baldini, D. Wei, K. Natesan Ramamurthy, M. Singh, and M. Yurochkin, "Your fairness may vary: Pretrained language model fairness in toxic text classification," in *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2245–2262. [Online]. Available: https://aclanthology.org/2022.findings-acl.176

[42] P. Delobelle, E. Tokpo, T. Calders, and B. Berendt, "Measuring fairness with biased rulers: A comparative study on bias metrics for pre-trained language models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association

for Computational Linguistics, Jul. 2022, pp. 1693–1706. [Online]. Available: https://aclanthology.org/2022.naacl-main.122

[43] Y. Wan and K.-W. Chang, "White men lead, black women help? benchmarking language agency social biases in llms," 2024. [Online]. Available: https://arxiv.org/abs/2404.10508