Proceedings of Machine Learning Research 275:1–25, 2025 4th Conference on Causal Learning and Reasoning

Counterfactual Token Generation in Large Language Models

Ivi Chatzi[*] ICHATZI@MPI-SWS.ORG
Nina Corvelo Benz[*] NINACOBE@MPI-SWS.ORG
Eleni Straitouri[*] ESTRAITOURI@MPI-SWS.ORG
Stratis Tsirtsis[*] STSIRTSIS@MPI-SWS.ORG
Manuel Gomez-Rodriguez MANUELGR@MPI-SWS.ORG
Max Planck Institute for Software Systems

Editors: Biwei Huang and Mathias Drton

Abstract

"Sure, I am happy to generate a story for you: Captain Lyra stood at the helm of her trusty ship, the Maelstrom's Fury, gazing out at the endless sea. [...] Lyra's eyes welled up with tears as she realized the bitter truth – she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself." Although this story, generated by a large language model, is captivating, one may wonder—how would the story have unfolded if the model had chosen "Captain Maeve" as the protagonist instead? We cannot know. State-of-the-art large language models are stateless—they maintain no internal memory or state. Given a prompt, they generate a sequence of tokens as an output using an autoregressive process. As a consequence, they cannot reason about counterfactual alternatives to tokens they have generated in the past. In this work, our goal is to enhance them with this functionality. To this end, we develop a causal model of token generation that builds upon the Gumbel-Max structural causal model. Our model allows any large language model to perform counterfactual token generation at almost no cost in comparison with vanilla token generation, it is embarrassingly simple to implement, and it does not require any fine-tuning nor prompt engineering. We implement our model on `Llama 3 8B-Instruct` and `Ministral-8B-Instruct`, and conduct a qualitative and a quantitative analysis of counterfactually generated text. We conclude with a demonstrative application of counterfactual token generation for bias detection, unveiling interesting insights about the model of the world constructed by large language models.

Keywords: Counterfactual reasoning, large language models, Gumbel-Max SCM

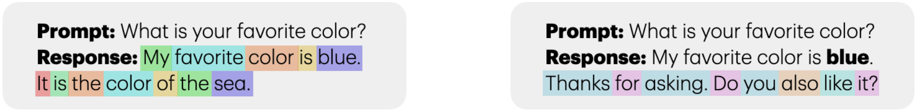
1. Introduction

Reasoning about "what might have been", about alternatives to our own past actions, is a landmark of human intelligence (Roese, 1997; Byrne, 2007; Van Hoeck et al., 2015). This type of reasoning, known as counterfactual reasoning, has been shown to play a significant role in the ability that humans have to learn from limited past experience and improve their decision making skills over time (Epstude and Roese, 2008; Markman et al., 2008; Roese and Epstude, 2017), it provides the basis for creativity and insight (Sternberg and Gastel, 1989), and it is tightly connected to the way we attribute causality and responsibility (Lagnado et al., 2013; Gerstenberg et al., 2021; Xiang et al., 2023; Tsirtsis et al., 2024). Can currently available large language models (LLMs) conduct counterfactual reasoning about alternatives to their own outputs? In this work, we argue that they cannot, by design.

---

[*] Authors contributed equally and are listed in alphabetical order.

© 2025 I. Chatzi, N.C. Benz, E. Straitouri, S. Tsirtsis & M. Gomez-Rodriguez.

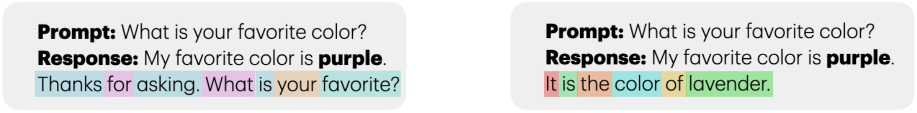


Prompt: What is your favorite color?
Response: My favorite color is blue.
It is the color of the sea.

Prompt: What is your favorite color?
Response: My favorite color is **blue**.
Thanks for asking. Do you also like it?

(*a*) Original generation

(*b*) Interventional generation with unmodified input



Prompt: What is your favorite color?
Response: My favorite color is **purple**.
Thanks for asking. What is your favorite?

Prompt: What is your favorite color?
Response: My favorite color is **purple**.
It is the color of lavender.

(*c*) Interventional generation with modified input

(*d*) Counterfactual generation with modified input

Figure 1: **Illustrative examples of autoregressive token generation.** In all panels, plain text indicates the input provided to the LLM and highlighted text indicates the output generated by the model. Each token in the output sequence is highlighted in a different color to represent the (stochastic) state of the sampler.[3] Panel (a) shows an LLM's output to a user's prompt using vanilla autoregressive token generation. Panels (b, c) show an LLM's output to an input comprising a user's prompt and an unmodified/modified part of the original output from Panel (a) using vanilla autoregressive token generation. Panel (d) shows an LLM's counterfactual output to an input comprising a user's prompt and a modified part of the output from Panel (a) using autoregressive token generation augmented with the Gumbel-Max SCM.

Currently available LLMs are stateless—they maintain no internal memory or state. Given an input prompt, they generate a sequence of tokens[1] as output using an autoregressive process (Bengio et al., 2000; Radford et al., 2019). At each time step, they first use a neural network to map the prompt and the (partial) sequence of tokens generated so far to a token distribution. Then, they use a sampler to draw the next token at random from the token distribution.[2] Finally, they append the next token to the (partial) sequence of tokens, and continue until a special end-of-sequence token is sampled. To understand why this autoregressive process is insufficient to reason counterfactually about alternatives to a previously generated sequence of tokens, we will use an illustrative example.

Consider that we ask an LLM to share its favorite color, as shown in Figure 1(*a*). Had the LLM chosen a different color (*e.g.*, purple instead of blue), what would the rest of its output have been? To answer such a counterfactual question, we need to implement two actions: (i) modify the (partial) sequence of tokens fed to the neural network used by the LLM and (ii) compel the sampler used by the LLM to *behave exactly as it did* in the original generation. Using currently available LLMs, we can readily implement the first action, which can be viewed as a causal intervention (Pearl, 2009; Peters et al., 2017). We just need to replace "blue" with "purple" in the (partial) sequence of tokens fed to the neural network. However, we cannot easily implement the second action, because the sampler does not specify how it *would have behaved* after taking the first action *while keeping everything else equal*. In fact, note that, if we provide the (modified) partial sequence up to and including the

---

[1] Tokens are the units that make up sentences and paragraphs. Examples of tokens include (sub-)words, symbols, numbers, and special end-of-sequence tokens.

[2] Multiple lines of evidence suggest that, if an LLM is forced to output tokens deterministically, its performance worsens (Holtzman et al., 2019).

[3] In Section 2, we formally define the state of the sampler as an exogenous noise variable in an SCM.

world “blue” (“purple”) as input to the LLM, there is no way to ensure that the LLM will generate an output that matches (the structure of) the original output, as shown in Figures 1(*a*), 1(*b*) and 1(*c*).[4]

Our contributions. Our key idea is to augment the autoregressive process of token generation underpinning an LLM, particularly the sampler used in the process, using a structural causal model (SCM) (Pearl, 2009). More specifically, we define the sampler through a causal mechanism that receives as input the distribution of the next token and a set of noise values, which determine the sampler’s (stochastic) state. Importantly, the use of a causal mechanism specifies how the sampler would have behaved under an intervention on the distribution of the next token and thus allows us to answer counterfactual questions about a previously generated sequence of tokens, as shown in Figure 1(*d*). Further, to instantiate our model, we use the Gumbel-Max SCM (Oberst and Sontag, 2019), an SCM shown to satisfy a desirable counterfactual stability property which, in the context of token generation, favors counterfactual output sequences that share similarities with the original sequence. Along the way, we also introduce an efficient implementation of the augmented autoregressive process that can generate counterfactual tokens at almost no cost in comparison with vanilla token generation. As a proof of concept, we implement our model on `Llama 3 8B-Instruct` and `Ministral-8B-Instruct`, and we conduct experiments to qualitatively and quantitatively analyze the similarity between an LLM’s original output and the one generated via counterfactual token generation. Additionally, we demonstrate the use of our methodology for bias detection, unveiling interesting insights about the model of the world constructed by large language models. We conclude with a comprehensive discussion of the limitations of our model, including additional avenues for applications.[5]

Further related work. Our work is closely related to a line of work on counterfactual text generation (Qin et al., 2019, 2020; Hao et al., 2021; Chen et al., 2022; Wang et al., 2024a,b; Nguyen et al., 2024b; Zellers et al., 2019; Li et al., 2023; Nguyen et al., 2024a; Gat et al., 2024). In this line of work, given pairs of factual statements and interventions over these statements, the goal is to generate counterfactual statements that match those made by humans—counterfactual statements that are consistent with the underlying model of the world shared by humans. To this end, existing methods typically fine-tune an LLM using a dataset comprising factual statements, interventions over these statements, and counterfactual statements made by humans. In contrast, in our work, our goal is to generate counterfactual statements that are consistent with the underlying model of the world constructed by a given LLM (Li et al., 2021, 2022; Patel and Pavlick, 2022; Vafa et al., 2024). In this context, our work also relates to a rapidly increasing number of empirical studies assessing the ability of LLMs to answer questions that require counterfactual reasoning (Frohberg and Binder, 2022; Jin et al., 2023; Kıcıman et al., 2023; Pawlowski et al., 2023; Jiang et al., 2023b; Betti et al., 2023; Liu et al., 2023; Miao et al., 2023; Wu et al., 2023; Nie et al., 2024; Ortu et al., 2024; Liu et al., 2024). Here, the LLMs are typically evaluated using multiple choice questions about a given set of factual and counterfactual statements. However, similarly as in the line of work on counterfactual text generation discussed previously, the counterfactual statements are made by humans.

The works by Ravfogel et al. (2025) and Bynum and Cho (2024), which have been undertaken concurrently and independently of our work, also use SCMs to model the autoregressive process underpinning LLMs as we do. The work by Ravfogel et al. (2025) is most closely related to ours;

---

[4] Note that using the same random seed is not sufficient because the number of tokens of the input in Figure 1(*a*) and the number of tokens of the inputs in Figures 1(*b*) and 1(*c*) differ.

[5] An open-source implementation of our model on `Llama 3 8B-Instruct` and `Ministral-8B-Instruct` is available at https://github.com/Networks-Learning/counterfactual-llms.

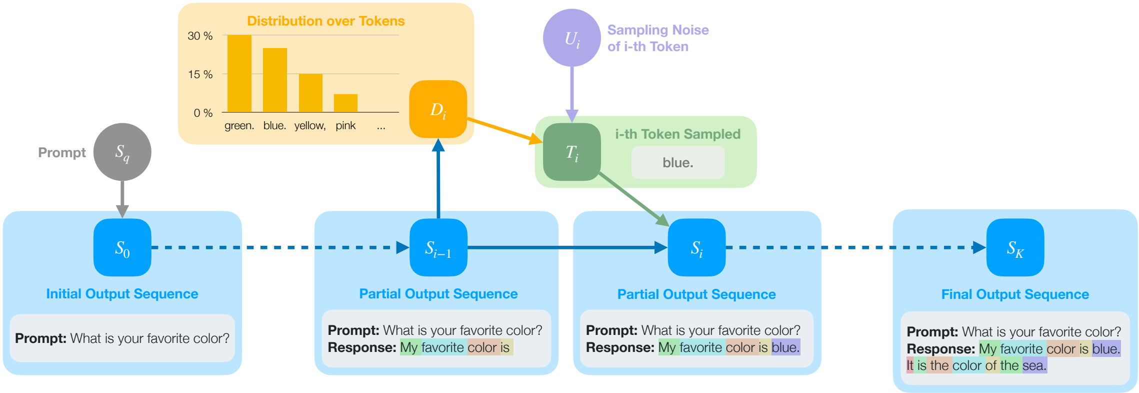

Distribution over Tokens
30 %
15 %
0 %
green.
blue.
yellow.
pink
...
D_i
U_i
Sampling Noise of i-th Token
T_i
i-th Token Sampled
blue.
Prompt
S_q
S_0
Initial Output Sequence
Prompt: What is your favorite color?
S_{i-1}
Partial Output Sequence
Prompt: What is your favorite color?
Response: My favorite color is
S_i
Partial Output Sequence
Prompt: What is your favorite color?
Response: My favorite color is blue.
S_K
Final Output Sequence
Prompt: What is your favorite color?
Response: My favorite color is blue.
It is the color of the sea.

Figure 2: **Causal graph of our proposed SCM $\mathcal{M}$ for token generation.** Boxes represent endogenous random variables and circles represent exogenous random (noise) variables. The value of each endogenous variable is given by a function of the values of its ancestors in the causal graph, as defined by Eq. 1. The value of each noise variable U_i is sampled independently from a given distribution P_U, and it determines the stochastic state of the LLM's sampler during the generation of token T_i (refer to Fig. 1).

they model autoregressive generation using the Gumbel-Max SCM and generate counterfactual strings to visualize and analyze the effects of interventions within (the network of) an LLM. However, in contrast to our work, they do not explicitly implement the sampler of an LLM as an SCM and use the same set of noise values during factual and counterfactual generation, but they sample the noise values using a posterior distribution inferred from the factual output. The work by Bynum and Cho (2024) is fundamentally different to ours. In the context of a specific task, they utilize an SCM to describe and quantify the causal relations between variables determined by the semantics of that specific task. In this context, it is also relevant to note that the Gumbel-Max SCM has previously been used to enable counterfactual reasoning in other domains such as Markov decision processes (Tsirtsis et al., 2021), temporal point processes (Noorbakhsh and Gomez-Rodriguez, 2022), and expert predictions (Benz and Gomez-Rodriguez, 2022).

2. A Causal Model of Token Generation

To formally express autoregressive token generation, we adopt (part of) the notation introduced by Duetting et al. (2024) in a different (non-causal) context. Let V denote the vocabulary (set) of tokens available to the LLM, which includes an end-of-sequence token $\perp$. Then, we denote by $V^* = V \cup V^2 \cup \cdots \cup V^K$ the set of sequences of tokens up to maximum length K, and by $\varnothing$ the empty token. An LLM takes as input a prompt sequence $s_q \in V^*$ and responds with an output sequence $s \in V^*$. The output sequence is generated using an autoregressive process. At each time step $i \in [K]$, the LLM first takes as input the concatenation of the prompt sequence s_q and the (partial) output sequence s_{i-1} and generates a distribution over tokens $d_i \in \Delta(V)$. Then, it samples the next token $t_i \sim d_i$ from the distribution d_i and creates the output sequence $s_i = s_{i-1} \circ t_i$, where $\circ$ denotes the concatenation of a token or sequence with another sequence. Further, if $t_i = \perp$, it terminates and returns $s = s_i$ and, otherwise, it continues to the next step $i + 1$ in the generation.

Given any prompt sequence, the above autoregressive process determines what (factual) output sequence the LLM generates as a response. However, given a generated output sequence, the above process does not determine what counterfactual output sequence the LLM would have generated if the prompt sequence, or some of the tokens in the output sequence, had been different. To address this limitation, we augment the autoregressive process using a structural causal model (SCM) (Pearl, 2009; Peters et al., 2017), which we denote as $\mathcal{M}$. Our SCM $\mathcal{M}$ is defined by the following assignments[6]:

$$S_0 = S_q, \quad D_i = \begin{cases} f_D(S_{i-1}) & \text{if } \texttt{last}(S_{i-1}) \neq \perp, \\ P_\varnothing & \text{otherwise} \end{cases}, \quad T_i = \begin{cases} f_T(D_i, U_i) & \text{if } D_i \neq P_\varnothing, \\ \varnothing & \text{otherwise} \end{cases}, \tag{1}$$

$$S_i = S_{i-1} \circ T_i \quad \text{and} \quad S = S_K,$$

where S_q and $\boldsymbol{U} = (U_i)_{i \in \{1,\ldots,K\}}$ are independent exogenous random variables, with $S_q \sim P_Q$ and $U_i \sim P_U$, respectively, f_D and f_T are given functions, $P_\varnothing$ is the point mass distribution on $\varnothing$, and $\texttt{last}(S_{i-1})$ denotes the last token of the sequence S_{i-1}. Here, the function f_D is defined by the architecture and parameters of the LLM and the choice of function f_T and distribution P_U determines the exact mechanism that the LLM's sampler uses to (stochastically) select the next token T_i. Note that, there always exists a pair of f_T and P_U such that the distribution D_i matches the distribution $P^{\mathcal{M}}(T_i)$ entailed by $\mathcal{M}$ (see Buesing et al. (2018), Lemma 2 for a technical argument). Moreover, note that, in the SCM $\mathcal{M}$, the output sequence S contains the prompt sequence to lighten the notation regarding interventions. For an illustration of the SCM and its causal graph, refer to Figure 2.

Under this augmented autoregressive process, given an output sequence $S = s$ and noise values $\boldsymbol{U} = \mathbf{u}$, we can generate the counterfactual output sequence the LLM would have generated if the prompt sequence, or some of the tokens in the output sequence had been different, deterministically. More formally, given an intervention $\text{do}[S_i = \tilde{s}]$, with $i \leq |s|$, the counterfactual output sequence $S = S_K$ can be computed recursively using the following expression:

$$S_j = \begin{cases} s_j & \text{if } j < i \\ \tilde{s} & \text{if } j = i \\ S_{j-1} \circ f_T(f_D(S_{j-1}), u_j) & \text{if } j > i \text{ and } \texttt{last}(S_{j-1}) \neq \perp \\ S_{j-1} & \text{otherwise.} \end{cases} \tag{2}$$

Note that the key element of this recursive expression for the counterfactual output sequence is the use of the same realized noise values u_j for $j \in [K]$ that were used to generate the factual output sequence s. However, without further assumptions, the counterfactual output sequence may be non-identifiable. This is because there may be multiple noise distributions P_U and functions f_T under which $P^{\mathcal{M}}(T_i) = D_i$, but each pair produces a different counterfactual output sequence—Oberst and Sontag (2019) make a similar argument in the context of Markov decision processes. In simpler terms, without explicitly modeling the stochastic mechanism by which the sampler selects the next token in the factual sequence, it is not possible to determine which tokens would have been selected in the counterfactual output sequence. In the next section, we address this issue by focusing on the class of Gumbel-Max SCMs to implement an LLM's sampler.

---

[6] We denote random variables with capital letters and realizations of random variables with lower case letters.

3. Counterfactual Token Generation Using Gumbel-Max SCMs

Under the class of Gumbel-Max SCMs, the function f_T that implements the sampling of the next token in the SCM $\mathcal{M}$ adopts the following functional form (Oberst and Sontag, 2019):

$$f_T(D_i, U_i) = \underset{t \in V}{\text{argmax}} \{\log D_{i,t} + U_{i,t}\}, \tag{3}$$

where $U_{i,t} \sim \text{Gumbel}(0,1)$ are independently distributed Gumbel variables that determine the stochastic state of the LLM's sampler during the generation of token T_i (refer to Fig. 1). Importantly, this class of SCMs has been shown to satisfy a desirable counterfactual stability property that can be intuitively expressed as follows. Assume that, at time step i, the augmented autoregressive process sampled token t_i given $d_i = f_D(s_{i-1})$. Then, in a counterfactual scenario where $D_i = d'$, it is *unlikely* that, at time step i, the augmented autoregressive process would have sampled a token t' other than t_i—the factual one—unless, under the token distribution d', the relative chance of generating token t_i decreased compared to other tokens. More formally, for any token distribution $d' \in \Delta(V)$ with $d' \neq d_i$ such that

$$\frac{P^{\mathcal{M}}(T_i = t_i \,|\, D_i = d')}{P^{\mathcal{M}}(T_i = t_i \,|\, D_i = d_i)} \geq \frac{P^{\mathcal{M}}(T_i = t' \,|\, D_i = d')}{P^{\mathcal{M}}(T_i = t' \,|\, D_i = d_i)},$$

it holds that, in the counterfactual scenario where $D_i = d'$, the counterfactual token $T_i \neq t'$.

To understand the intuition behind counterfactual stability, consider the following example. Assume that the vocabulary V contains 2 tokens "A" and "B", the (factual) distribution d_i assigns values 0.6 and 0.4, respectively, and the Gumbel-Max SCM samples token "A". Consider an intervention that changes d_i to a distribution d' that assigns values 0.7 and 0.3 to "A" and "B", respectively. Then, the property of counterfactual stability ensures that, during the counterfactual generation, token "A" is also sampled because, in comparison to the factual generation, the relative odds are higher, *i.e.*, 0.7 to 0.3 vs. 0.6 to 0.4. In a way, counterfactual stability "prioritizes" the token sampled in the factual generation, maintaining consistency between the factual and counterfactual text. For a further discussion of the stability property and alternatives to the Gumbel-Max SCM, refer to Section 5 .

In practice, in addition to solving the non-identifiability issues discussed previously, the use of Gumbel-Max SCMs allows for an efficient procedure to sample a sequence of counterfactual tokens with minimal additional memory requirements compared to vanilla token generation. We summarize the procedure in Algorithm 1. Recall that, to generate the counterfactual output sequence, one needs to use the same values u_j for the noise variables that were used during the factual generation and then perform an autoregressive computation based on Equation 2. Instead of storing the values u_j for all time steps $j \in [K]$, whose dimensionality matches the size of the vocabulary V, Algorithm 1 employs a simple idea: it stores the state of the random number generator r_j used at each time step $j \in [K]$ of the factual generation. Then, during the counterfactual generation, it regenerates the values $u_j = \text{GenGumbel}(r_j)$ on the fly.[7]

Remarks on implementation aspects of LLMs. In practice, to avoid sampling tokens with very low probability, LLMs may not directly sample from the distribution over tokens d_i at each time step i. Instead, a common practice is to sample from a distribution $\hat{d}_i \in \Delta(V_i)$, where $\hat{d}_{i,t} \propto d_{i,t}$

---

[7] Storing the realized values of the Gumbel variables requires storing $\mathcal{O}(KV)$ float values since $u_j \in \mathbb{R}^V$. On the other hand, the states of random number generators r_j take values in $\mathbb{N}^d$, where, for instance, $d = 16$ in `pytorch` (Paszke et al., 2019). Thus, our approach requires $\mathcal{O}(K)$ additional integer memory compared to vanilla token generation.

ALGORITHM 1: It returns a counterfactual sequence of tokens using a Gumbel-Max SCM

```
Input: Random number generator states $\mathbf{r}$, factual output sequence $s$, intervention $(i, \tilde{s})$.
Output: Counterfactual output sequence $s'$.
for $j = 1, \ldots, K$ do
    if $j < i$ then
        $s'_j = s_j$
    else if $j = i$ then
        $s'_j = \tilde{s}$
    else if $j > i \wedge \texttt{last}(s'_{j-1}) \neq \perp$ then
        $u_j = \text{GenGumbel}(r_j)$
        $d'_{j,t} = f_D(s'_{j-1})$
        $t_j = \text{argmax}_{t \in V}\{\log d'_{j,t} + u_{j,t}\}$
        $s'_j = s'_{j-1} \circ t_j$
    else
        $s'_j = s'_{j-1}$
    end
end
Return $s'_K$
```

if $t \in V_i$ and $\hat{d}_{i,t} = 0$ otherwise, where V_i is either the set of most likely tokens of size k under d_i—known as “top-k” sampling—or the set of most likely tokens whose cumulative probability exceeds a given value p under d_i—known as “top-p” or “nucleus” sampling (Holtzman et al., 2019). We can readily implement top-k sampling and top-p sampling in the SCM $\mathcal{M}$ by restricting the argmax in Equation 3 to the respective set V_i. However, in general, the resulting model is not guaranteed to satisfy counterfactual stability.

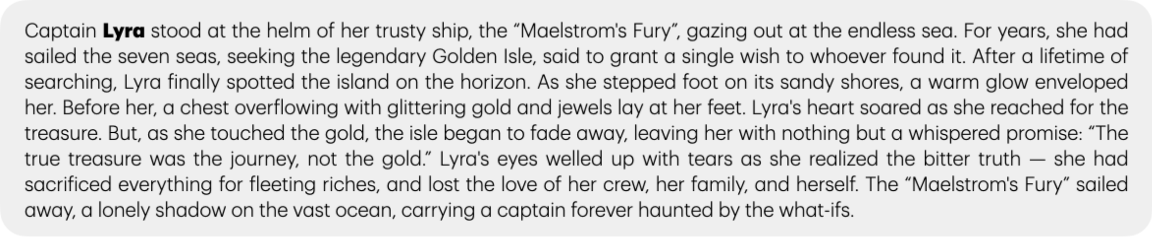
In all state-of-the-art LLMs, to ensure that the distribution d_i over tokens at each time step i is a valid probability distribution, the final layer in their neural network is a softmax layer. A crucial feature of this layer is the *temperature* parameter, τ, which controls the level of uncertainty in d_i. Intuitively, higher values of τ result in a more uniform distribution, while as τ approaches zero, the distribution concentrates increasingly on the most probable next token. In the next section, we perform a series of experiments in which we analyze the performance of counterfactual token generation, examining the effects of varying temperature values, as well as the application of top-k and top-p sampling.

4. Experiments

In this section, we experiment with an implementation of our model on `Llama 3 8B-Instruct` (Dubey et al., 2024) and `Ministral-8B-Instruct` (Jiang et al., 2023a), two popular open-weights large language models. We start by qualitatively analyzing an example of counterfactual story generation. Next, we quantitatively analyze the similarity between factual and counterfactual text. We conclude with an application of counterfactual token generation in detecting model biases towards demographic groups.[8]

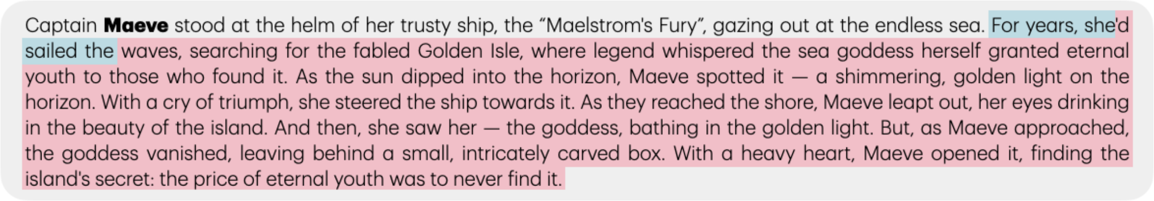
---

[8] All experiments ran on an cluster of machines, each equipped with 24 Intel(R) Xeon(R) 3GHz CPU cores, 1024GBs of memory and 2 NVIDIA A100 80GB GPUs.



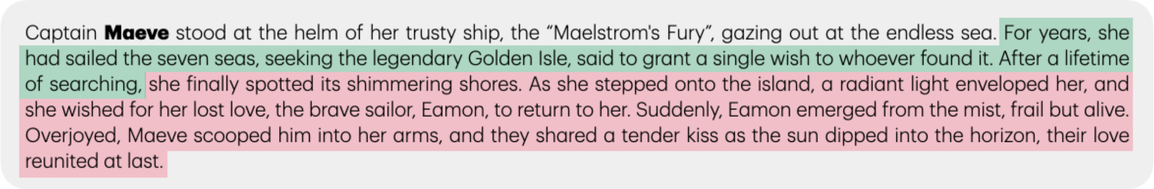
Captain **Lyra** stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(a) Factual story



Captain **Maeve** stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she'd sailed the waves, searching for the fabled Golden Isle, where legend whispered the sea goddess herself granted eternal youth to those who found it. As the sun dipped into the horizon, Maeve spotted it — a shimmering, golden light on the horizon. With a cry of triumph, she steered the ship towards it. As they reached the shore, Maeve leapt out, her eyes drinking in the beauty of the island. And then, she saw her — the goddess, bathing in the golden light. But, as Maeve approached, the goddess vanished, leaving behind a small, intricately carved box. With a heavy heart, Maeve opened it, finding the island's secret: the price of eternal youth was to never find it.

(b) Story generated with interventional token generation



Captain **Maeve** stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, she finally spotted its shimmering shores. As she stepped onto the island, a radiant light enveloped her, and she wished for her lost love, the brave sailor, Eamon, to return to her. Suddenly, Eamon emerged from the mist, frail but alive. Overjoyed, Maeve scooped him into her arms, and they shared a tender kiss as the sun dipped into the horizon, their love reunited at last.

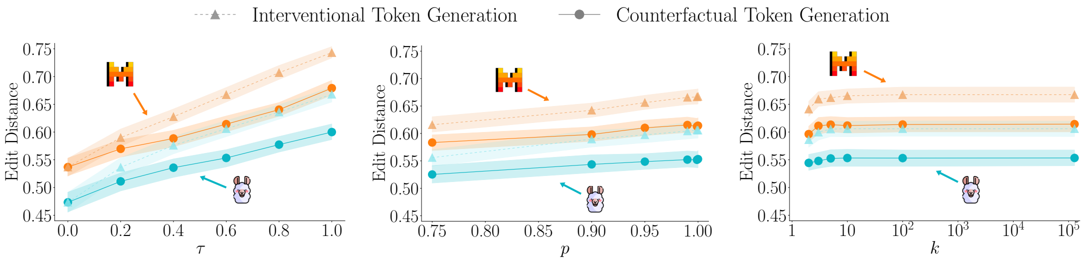
(c) Story generated with counterfactual token generation

Figure 3: **Examples of factual, interventional and counterfactual stories.** Panel (a) shows a factual story, as given by the LLM. Panels (b) and (c) show variants of the story resulting from interventional and counterfactual token generation, respectively. In panels (b), (c), we give as input to the LLM the original prompt along with the first sentence of the factual output (non-highlighted text), modified by replacing "Lyra" with "Maeve". Blue (green)-highlighted text indicates the tokens of the output that are identical in the factual story and its interventional (counterfactual) counterpart. Red-highlighted text indicates the differences. In both panels, the temperature parameter is set to $\tau = 0.9$.

4.1. How would the story have unfolded for "Captain Maeve"?

As discussed in Section 3, by using the Gumbel-Max SCM, our approach to counterfactual token generation is guaranteed to satisfy the property of counterfactual stability—counterfactual token generation "prioritizes" selecting the same tokens T_i that were selected during the factual generation. As a consequence, we expect the counterfactual text generated using counterfactual token generation to be similar to the factual text. Here, we investigate this qualitatively through an anecdotal example of story generation.

We use the implementation of our model on `Llama 3 8B-Instruct` with the system prompt *"Be creative and keep your response as short as possible."* and a query prompt *"Tell me a fantasy story about a captain. The story should have either a happy or a sad ending."* Figure 3(a) shows the (factual) generated story about Captain Lyra, her ship the Maelstrom's Fury, and her quest to find a treasure on the Golden Isle. Then, we use the original prompt along with part of the factual output (*i.e.*, the first sentence of the story) as input to the model, modifying the protagonist's name from "Lyra" to "Maeve", and we regenerate the rest of the output using two approaches:



(a) Gumbel-Max SCM (b) Top-p Gumbel-Max SCM (c) Top-k Gumbel-Max SCM

Figure 4: **Comparison between interventional and counterfactual token generation.** The panels show the edit distance between the factual token sequence and the sequence generated by interventional and counterfactual token generation using (a) the Gumbel-Max SCM defined in Equation 3, (b) the top-p Gumbel-Max SCM, and (c) the top-k Gumbel-Max SCM discussed at the end of Section 3, against various values of the temperature parameter τ, p and k, respectively. In panels (b, c) the temperature parameter is set to $\tau = 0.6$. In all three panels, the edit distance is averaged over 4,000 output sequences, resulting from two independent interventions per factual sequence, and shaded areas represent 95% confidence intervals. The icons 🦙 and 🟧 indicate results for `Llama 3 8B-Instruct` and `Ministral-8B-Instruct` respectively.

1. **Interventional token generation (see Fig 1(*c*)):** it regenerates the second part of the output using vanilla autoregressive token generation, *i.e.*, it samples new noise values u_j for the second part of the output using different states r_j for the random number generator from those used in the factual generation.
2. **Counterfactual token generation (see Fig 1(*d*)):** it regenerates the second part of the output using Algorithm 1, *i.e.*, it reuses the same states r_j for the random number generator and, hence, the same noise values u_j as the ones used in the factual generation.

Figures 3(*b*), 3(*c*) present two alternative versions of the factual story generated using the methods mentioned above. These stories reveal several interesting insights. The story generated with interventional token generation starts diverging from the factual story after only a few tokens, as the method lacks memory of the noise values u_j that resulted in the original output. In contrast, the initial part of the counterfactual output remains identical to the factual output, as expected, due to the counterfactual stability property of the Gumbel-Max SCM and the minor nature of changing the protagonist's name. Although one may expect this to apply for the rest of the counterfactual output, thinking that the protagonist's name would be irrelevant to the narrative of this particular story, this is not the case. Perhaps surprisingly, the use of "Maeve" instead of "Lyra" results in a partially different output, illustrating that the LLM's probability distributions over next tokens are sensitive even to minor changes. In Appendix A, we also observe differences between the factual and counterfactual outputs resulting from other seemingly irrelevant interventions, such as changing the name of the ship, removing the adjective "trusty" or replacing the word "sea" with "blue".

4.2. How similar is counterfactually generated text to the factual one?

In the previous section, we demonstrated through an example that counterfactual token generation results in text that is (partially) similar to the factual text, as expected due to the property of counterfactual stability. Here, we empirically verify this expectation using a quantitative analysis and explore how it is affected by the model parameters.

Experimental setup. We first use the implementation of our model on `Llama 3 8B-Instruct` and `Ministral-8B-Instruct` to generate (factual) outputs to 2,000 question prompts sourced from the LMSYS Chat 1M dataset (Zheng et al., 2023). As a system prompt we use *"Keep your replies short and to the point."*. Further, for each factual output, we perform two interventions where we replace a randomly selected token t_i with a token $t' \neq t_i$.[9] One of the two interventions restricts the choice of t_i to the first half of the output sequence and the other restricts it to the second half. Then, for each intervened factual output, we feed the concatenation of the question prompt and the first part of the intervened factual output up to and including token t' as input to our model. We regenerate the second part of the output after token t' using (i) interventional token generation and (ii) counterfactual token generation, as described in Section 4.1. Finally, we measure the lexicographic similarity between the regenerated second part of the output and its factual counterpart using their (normalized) Levenshtein edit distance (Levenshtein, 1966). In our experiments, we implement our model using the Gumbel-Max SCM defined in Equation 3 as well as the top-p Gumbel-Max SCM and top-k Gumbel-Max SCM discussed at the end of Section 3.

Results. Figure 4 summarizes the results, which show that, across different SCMs and LLMs, the output sequences generated using counterfactual token generation are more similar to the factual sequences (*i.e.*, the edit distance is lower) than the output sequences generated using interventional token generation. This suggests that, even though the top-p and top-k Gumbel-Max SCMs are not guaranteed to satisfy counterfactual stability, in practice, counterfactual token generation under both SCMs does "prioritize" selecting the same tokens T_i that were selected during the factual generation. Although this pattern is consistent across the two LLMs, we observe that output sequences generated by both interventional and counterfactual token generation are more similar to their respective factual sequences for `Llama 3 8B-Instruct` than `Ministral-8B-Instruct`.

4.3. Does counterfactual token generation reveal model biases?

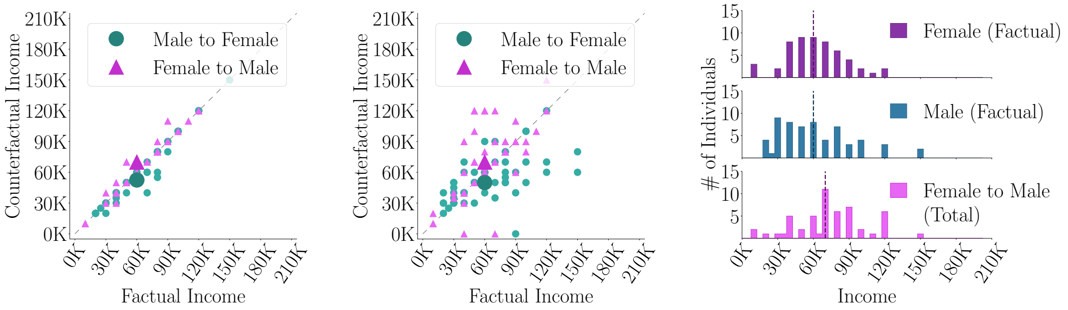
Common approaches to addressing questions of bias and fairness rely on making counterfactual comparisons based on sensitive attributes (Kusner et al., 2017). For example, would a person's income have been the same if their race or sex were different? In this section, we focus on a census data generation task, and demonstrate the use of counterfactual token generation to investigate potential biases of an LLM towards demographic groups.

Experimental setup. We first use the implementation of our model on `Llama 3 8B-Instruct` and `Ministral-8B-Instruct` to generate (factual) census data. For each model, we use the same input prompt three times with different seeds (see Appendix B for details), requesting 50 individuals each time. The attributes of generated individuals include their age, sex, citizenship, race, ethnicity, marital status, number of children, occupation, income, and education, in this given order, and we discard individuals for whom the generated income is exactly zero. The factual data generated by the models result in a corpus of 114 and 158 fictional individuals for `Llama 3 8B-Instruct` and `Ministral-8B-Instruct`, respectively.

For each fictional person, we consider all possible interventions on each of the sensitive attributes of sex and race. For each intervention on sex and race, we concatenate the input prompt with the

---

[9] To select t', we set the probability of t_i in d_i to 0, re-scale the values of d_i and use top-p sampling with $p = 0.9$.



(*a*) Change in income upon intervention on sex (direct effect)

(*b*) Change in income upon intervention on sex (total effect)

(*c*) Distribution of factual and counterfactual income

Figure 5: **Comparison between factual and counterfactual income.** Panel (a) shows the change in income of male (female) individuals had they been female (male), while keeping fixed the rest of their attributes preceding income in the output sequence. Panel (b) shows the change in income of male (female) individuals had they been female (male), while keeping fixed the attributes preceding sex but allowing the attributes between sex and income to change in the output sequence. Panel (c) shows the factual distributions of income of female and male individuals and the counterfactual distribution of income of female individuals under the same intervention as in panel (b). Enlarged points in panels (a, b) and dashed lines in panel (c) correspond to the median income. In all panels, we use `Llama 3 8B-Instruct` and set the temperature parameter to $\tau = 0.8$.

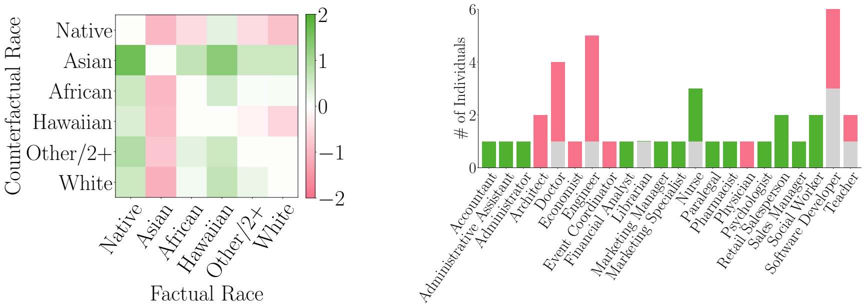
initial part of the output that includes the fictional person's description up to and including the intervened sensitive attribute. This concatenated input is then used by our model to regenerate the latter part of the output, following the intervention, using counterfactual token generation (*i.e.*, Algorithm 1). Then, we compare the factual and counterfactual values of income, education and occupation to measure the *total effect* of sex and race on those attributes. In addition, for each intervention on sex, we also measure the *direct effect* of sex on income. To this end, we concatenate the input prompt with the initial part of the output that includes the intervened value of sex and the factual values of all other attributes preceding income. This concatenated input is then used by our model to regenerate the latter part of the output starting from the income attribute.[10]

Results. Figure 5 summarizes the results with respect to the effect of sex on income using `Llama 3 8B-Instruct`. For both the direct and the total effect of sex on income, Figures 5(*a*) and 5(*b*) show that the generated income for most male (female) individuals would have remained unchanged or decreased (increased) had they been female (male), however, the total effect exhibits larger variance. This suggests that, in the LLM's world model, both the direct and total effects of sex on income are present, but at a moderate level. Interestingly, the (potentially biased) effect of sex on income cannot be identified solely from the factual distributions of female and male income, which present exactly the same median, as shown in Figure 5(*c*). For additional results using `Llama 3 8B-Instruct` and `Ministral-8B-Instruct`, refer to Appendix C.

Figure 6 summarizes the results with respect to the effect of race on education and occupation using `Ministral-8B-Instruct`. Figure 6(*a*) shows that, for individuals of all (generated) races, there exists at least one other race that, had they belonged to it, they would have experienced

---

[10] For further details regarding the difference between total and direct effects, refer to Pearl (2009), Chapter 4.



(a) Change in education level upon intervention on race

(b) Change in occupation upon intervention on race

Figure 6: **Comparison between factual and counterfactual education and occupation.** Panel (a) shows the average difference in the education level of individuals of each race had their race been different. Here, positive values indicate an improvement in education and negative values indicate a decline. Panel (b) shows the distribution shift of occupations among Asian American individuals had they been Black or African American. Green (red) sections indicate the counterfactual increase (decrease) in the number of individuals that practice each occupation. In both panels, we use `Ministral-8B-Instruct` and set the temperature parameter to $\tau = 0.8$.

a significant increase or decrease in their education level (refer to Appendix B for the assignment of each education level to a numerical value and for the factual distribution of individuals across races). Specifically, we observe a consistent pattern in which the LLM would have increased the level of education for (i) American Indian or Alaska Native (Native) and (ii) Native Hawaiian or Other Pacific Islander (Hawaiian) individuals had they belonged to any other race. In contrast, it would have decreased the level of education for Asian American (Asian) individuals.[11] Figure 6(b) shows that, for Asian American individuals, their occupation would have shifted from STEM to humanities-related occupations had they been Black or African American. In Appendix C, we present qualitatively similar results using `Llama 3 8B-Instruct`.

5. Discussion and Limitations

In this section, we discuss several assumptions and limitations of our work, pointing out avenues for future research.

Methodology. Our causal model of the autoregressive process underpinning large language models operationalizes the counterfactual stability property using the Gumbel-Max SCM. It would be interesting to understand the sensitivity of counterfactual token generation to this specific choice of SCM and implement counterfactual token generation using alternative SCMs obeying counterfactual stability (Lorberbom et al., 2021; Haugh and Singal, 2023). In this context, it is also important to acknowledge that, while counterfactual stability is somewhat an appealing property, Haugh and Singal (2023) have recently argued that its appropriateness depends on the application and should be justified by domain specific knowledge. Moreover, they have shown that there are cases where counterfactual stability may permit counterfactuals that it was designed to exclude. Motivated by this

---

[11] The terms for races in parentheses and in Figure 6(a) are shortened descriptions, which we use for brevity. Refer to Appendix B for a list of all official descriptions.

observation, in Appendix D, we include additional experiments on counterfactual token generation using a classical sampler for categorical distributions, which can be viewed as an SCM that does not satisfy counterfactual stability.

Further, there are reasons to believe that counterfactual statements generated using our model may be inconsistent with the underlying causal model of the world shared by humans (Li et al., 2021, 2022; Patel and Pavlick, 2022; Vafa et al., 2024) and this may render them unreliable for certain uses cases. An interesting future direction would be to explore the use of our methodology in conjunction with human feedback to train (or fine-tune) LLMs that better understand causal relationships.

Applications. Counterfactual token generation may be useful for understanding the inner workings of an LLM. For example, it may be used as a tool for evaluating causal dependencies among physical or social attributes within the world model learned by an LLM, as in our bias experiments in Section 4.3, or quantifying the "importance" of different parts of a model's output in reaching a final conclusion, similarly as in feature attribution (Ribeiro et al., 2016; Adebayo et al., 2018). Further, counterfactual token generation may also be useful for creating novel interfaces for LLM-human collaborations. For example, it could be used in scenarios in which a user is satisfied with a portion of an LLM's output but wishes to modify a few words while maintaining similarity between the subsequent text and the original output. Additionally, counterfactual token generation may be proven useful in (applications in) linguistics and cognitive science.

Evaluation. We have implemented our model on two LLMs, namely `Llama 3 8B-Instruct` and `Ministral-8B-Instruct`. It would be useful to implement our model on other LLMs and use counterfactual token generation to reveal similarities and differences between the underlying models of the world constructed by different LLMs. In this context, it would be insightful to see whether the sensitivity of an LLM's counterfactual output changes as its number of parameters increases.

6. Conclusions

In this work, we have proposed a causal model of token generation. Using the Gumbel-Max SCM, we have introduced a methodology that enhances state-of-the-art LLMs with the ability to perform counterfactual token generation, allowing them to reason about past alternatives to their own outputs. We have experimentally analyzed the similarity between an LLM's original output and the one generated by counterfactual token generation, and we have demonstrated the use of our methodology in bias detection.

Acknowledgments

Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 945719).

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Nina L Corvelo Benz and Manuel Gomez Gomez-Rodriguez. Counterfactual inference of second opinions. In *Uncertainty in Artificial Intelligence*, pages 453–463. PMLR, 2022.

Lorenzo Betti, Carlo Abrate, Francesco Bonchi, and Andreas Kaltenbrunner. Relevance-based infilling for natural language counterfactuals. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 88–98, 2023.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *Proceedings of the 7th International Conference on Learning Representations*, 2018.

Lucius E. J. Bynum and Kyunghyun Cho. Language models as causal effect generators. *arXiv preprint arXiv:2411.08019*, 2024.

Ruth MJ Byrne. Precis of the rational imagination: How people create alternatives to reality. *Behavioral and Brain Sciences*, 30(5-6):439–453, 2007.

Jiangjie Chen, Chun Gan, Sijie Cheng, Hao Zhou, Yanghua Xiao, and Lei Li. Unsupervised editing for counterfactual stories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10473–10481, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 144–155, 2024.

Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking. *Personality and social psychology review*, 12(2):168–192, 2008.

Jörg Frohberg and Frank Binder. Crass: A novel data set and benchmark to test counterfactual reasoning of large language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2126–2140, 2022.

Yair Ori Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. Faithful explanations of black-box NLP models using LLM-generated counterfactuals. In *International Conference on Learning Representations*, 2024.

Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological review*, 128(5):936, 2021.

Changying Hao, Liang Pang, Yanyan Lan, Yan Wang, Jiafeng Guo, and Xueqi Cheng. Sketch and customize: A counterfactual story generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12955–12962, 2021.

Martin B Haugh and Raghav Singal. Counterfactual analysis in dynamic latent state models. In *International Conference on Machine Learning*, pages 12647–12677. PMLR, 2023.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Haitao Jiang, Lin Ge, Yuhe Gao, Jianian Wang, and Rui Song. Large language model for causal decision making. *arXiv preprint arXiv:2312.17122*, 2023b.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in Neural Information Processing Systems*, 2017.

David A Lagnado, Tobias Gerstenberg, and Ro'i Zultan. Causal responsibility and counterfactuals. *Cognitive science*, 37(6):1036–1073, 2013.

V Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*, 1966.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. Prompting large language models for counterfactual generation: An empirical study. *arXiv preprint arXiv:2305.14791*, 2023.

Xiao Liu, Da Yin, Chen Zhang, Yansong Feng, and Dongyan Zhao. The magic of if: Investigating causal reasoning abilities in large language models of code. *arXiv preprint arXiv:2305.19213*, 2023.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*, 2024.

Guy Lorberbom, Daniel D Johnson, Chris J Maddison, Daniel Tarlow, and Tamir Hazan. Learning generalized gumbel-max causal mechanisms. In *Advances in Neural Information Processing Systems*, volume 34, pages 26792–26803, 2021.

Keith D Markman, Matthew N McMullen, and Ronald A Elizaga. Counterfactual thinking, persistence, and performance: A test of the reflection and evaluation model. *Journal of Experimental Social Psychology*, 44(2):421–428, 2008.

Xin Miao, Yongqi Li, and Tieyun Qian. Generating commonsense counterfactuals for stable relation extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5654–5668, 2023.

Van Bach Nguyen, Jörg Schlötterer, and Christin Seifert. Ceval: A benchmark for evaluating counterfactual text generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 55–69, 2024a.

Van Bach Nguyen, Paul Youssef, Jörg Schlötterer, and Christin Seifert. Llms for generating and evaluating counterfactuals: A comprehensive study. *arXiv preprint arXiv:2405.00722*, 2024b.

Allen Nie, Yuhui Zhang, Atharva Shailesh Amdekar, Chris Piech, Tatsunori B Hashimoto, and Tobias Gerstenberg. Moca: Measuring human-language model alignment on causal and moral judgment tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Kimia Noorbakhsh and Manuel Gomez-Rodriguez. Counterfactual temporal point processes. *Advances in Neural Information Processing Systems*, 35:24810–24823, 2022.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. *arXiv preprint arXiv:2402.11655*, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International conference on learning representations*, 2022.

Nick Pawlowski, James Vaughan, Joel Jennings, and Cheng Zhang. Answering causal questions with augmented llms. 2023.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053, 2019.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Shauli Ravfogel, Anej Svete, Vésteinn Snæbjarnarson, and Ryan Cotterell. True counterfactual generation from language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Neal J Roese. Counterfactual thinking. *Psychological bulletin*, 121(1):133, 1997.

Neal J Roese and Kai Epstude. The functional theory of counterfactual thinking: New evidence, new challenges, new insights. In *Advances in experimental social psychology*, volume 56, pages 1–79. Elsevier, 2017.

Robert J Sternberg and Joyce Gastel. If dancers ate their shoes: Inductive reasoning with factual and counterfactual premises. *Memory & Cognition*, 17:1–10, 1989.

Stratis Tsirtsis, Abir De, and Manuel Gomez-Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. *Advances in Neural Information Processing Systems*, 34: 30127–30139, 2021.

Stratis Tsirtsis, Manuel Gomez Rodriguez, and Tobias Gerstenberg. Towards a computational model of responsibility judgments in sequential human-ai collaboration. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, 2024.

Keyon Vafa, Justin Y Chen, Jon Kleinberg, Sendhil Mullainathan, and Ashesh Rambachan. Evaluating the world model implicit in a generative model. In *Advances on Neural Information Processing Systems*, 2024.

Nicole Van Hoeck, Patrick D Watson, and Aron K Barbey. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in human neuroscience*, 9:420, 2015.

Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. A survey on natural language counterfactual generation. *arXiv preprint arXiv:2407.03993*, 2024a.

Ziao Wang, Xiaofeng Zhang, and Hongwei Du. Beyond what if: Advancing counterfactual text generation with structural causal modeling. In *IJCAI*, 2024b.

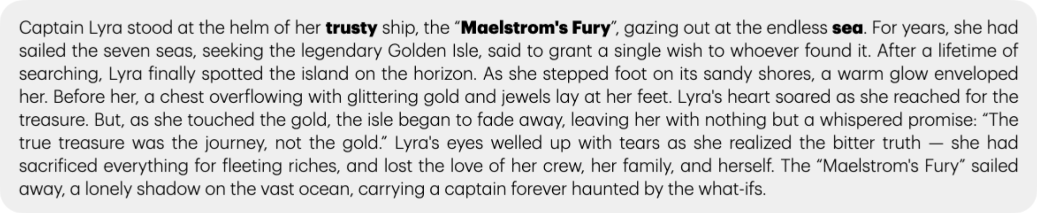
Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*, 2023.

Yang Xiang, Jenna Landy, Fiery A Cushman, Natalia Vélez, and Samuel J Gershman. Actual and counterfactual effort contribute to responsibility attributions in collaborative tasks. *Cognition*, 241: 105609, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

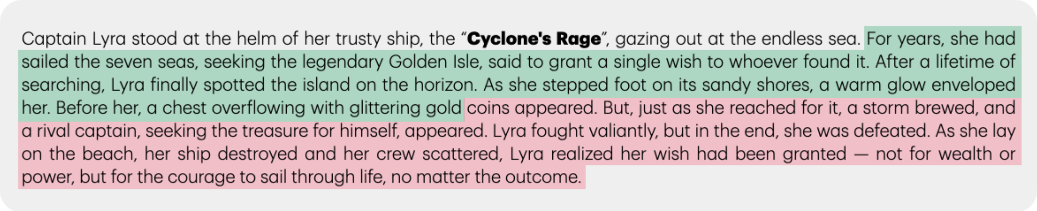
Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023.

Appendix A. Additional counterfactual stories



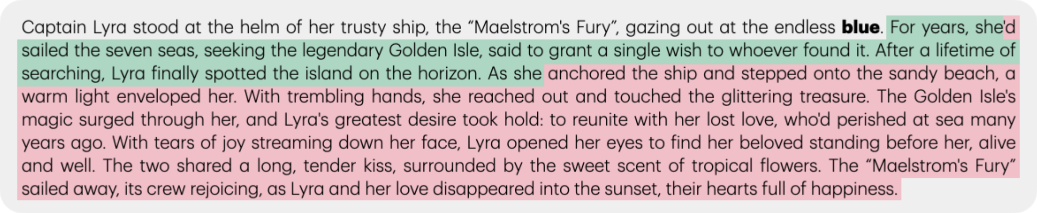
Captain Lyra stood at the helm of her **trusty** ship, the "**Maelstrom's Fury**", gazing out at the endless **sea**. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever haunted by the what-ifs.

(*a*) Factual story



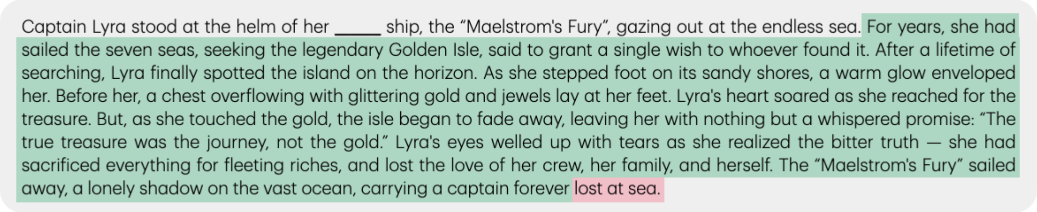
Captain Lyra stood at the helm of her trusty ship, the "**Cyclone's Rage**", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold coins appeared. But, just as she reached for it, a storm brewed, and a rival captain, seeking the treasure for himself, appeared. Lyra fought valiantly, but in the end, she was defeated. As she lay on the beach, her ship destroyed and her crew scattered, Lyra realized her wish had been granted — not for wealth or power, but for the courage to sail through life, no matter the outcome.

(*b*) Story generated with counterfactual token generation, after modifying the ship's name



Captain Lyra stood at the helm of her trusty ship, the "Maelstrom's Fury", gazing out at the endless **blue**. For years, she'd sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she anchored the ship and stepped onto the sandy beach, a warm light enveloped her. With trembling hands, she reached out and touched the glittering treasure. The Golden Isle's magic surged through her, and Lyra's greatest desire took hold: to reunite with her lost love, who'd perished at sea many years ago. With tears of joy streaming down her face, Lyra opened her eyes to find her beloved standing before her, alive and well. The two shared a long, tender kiss, surrounded by the sweet scent of tropical flowers. The "Maelstrom's Fury" sailed away, its crew rejoicing, as Lyra and her love disappeared into the sunset, their hearts full of happiness.

(*c*) Story generated with counterfactual token generation, after changing "sea" to "blue"



Captain Lyra stood at the helm of her _____ ship, the "Maelstrom's Fury", gazing out at the endless sea. For years, she had sailed the seven seas, seeking the legendary Golden Isle, said to grant a single wish to whoever found it. After a lifetime of searching, Lyra finally spotted the island on the horizon. As she stepped foot on its sandy shores, a warm glow enveloped her. Before her, a chest overflowing with glittering gold and jewels lay at her feet. Lyra's heart soared as she reached for the treasure. But, as she touched the gold, the isle began to fade away, leaving her with nothing but a whispered promise: "The true treasure was the journey, not the gold." Lyra's eyes welled up with tears as she realized the bitter truth — she had sacrificed everything for fleeting riches, and lost the love of her crew, her family, and herself. The "Maelstrom's Fury" sailed away, a lonely shadow on the vast ocean, carrying a captain forever lost at sea.

(*d*) Story generated with counterfactual token generation, after deleting the word "trusty"

Figure 7: **Comparison between the factual story and counterfactual variants.** Panel (a) shows the same factual story as in Section 4.1. Panels (b, c, d) show the story resulting from various interventions. In each case, the first sentence (non-highlighted text) is provided as input to the LLM, with the word(s) in bold (or left empty) representing the intervention. The remainder of the output is regenerated using counterfactual token generation. Text highlighted in green indicates the tokens of the output that are identical in the factual story and its counterfactual counterpart. Red-highlighted text indicates the differences. In all panels, the temperature parameter is set to $\tau = 0.9$.

Appendix B. Additional details on the experimental setup of Section 4.3

In this section, we provide additional details about the census generation experiment discussed in Section 4.3.

Data generation. Figure 8 shows the complete system and user prompts used to generate the census data. For race and ethnicity, we instructed the two LLMs, through the system prompt, to select values among those reported in the latest (2020) US Census. We used this prompt three times with different seeds. Despite our request for 50 individuals per generation, `Llama 3 8B-Instruct` only generated 45, 34 and 48 individuals each time, resulting in a total of 127 individuals, and `Ministral-8B-Instruct` generated 72, 42 and 44 individuals each time, resulting in a total of 158 individuals. The census data for `Llama 3 8B-Instruct` included individuals for whom the generated income is exactly zero, which we discarded, resulting in a total of 114 individuals. Additionally, for some interventions, the models generated incomplete counterfactual outputs; we excluded those cases from further analysis. Table 3 and figure 9 show the factual distributions of sex and race for the individuals generated by each model.

> **System:** Return only the following information: Age, Sex, Citizenship, Race, Ethnicity, Marital Status, Number of Children, Occupation, Income, Education. For Race, choose only between following options: White American, Black or African American, American Indian or Alaska Native, Asian American, Native Hawaiian or Other Pacific Islander, Other or Two or more races (multiracial). For Ethnicity, choose only between following options: Non-Hispanic/Latino or Hispanic/Latino. Return a list in json format delimited by “```”.
> **User:** Generate census data of 50 fictional people.

Figure 8: The prompt used for census data generation.

Description of race values and education level. Table 1 contains the full description of each race attribute value, of which shortened versions were used in Figure 6(*a*). Table 2 lists the numerical values assigned to the (categorical) education attribute values, which were used to compute the difference in education level shown in Figure 6(*a*).

Short	Full
Native	American Indian or Alaska Native
Asian	Asian American
African	Black or African American
Hawaiian	Native Hawaiian or Other Pacific Islander
Other/2+	Other or Two or more races (multiracial)
White	White American

Table 1: Short and full description of all races

Education	Numerical Value
High school diploma, High school, High School Diploma, High School	1
Associate's degree, Associate's Degree, Associate degree, Associate Degree, Associate's, Associate, Undergraduate, Some college, Some College, College, Vocational Training	2
Bachelor's degree, Bachelor's Degree, Bachelor's, Nursing Degree	3
Master's degree, Master's Degree, Master's	4
Ph.D., PhD, Doctorate degree, Doctorate Degree, Doctorate, Doctoral degree, Doctoral Degree, JD, Juris Doctor, Juris Doctor (JD), Law degree, Law degree, PharmD, Pharmacy Degree, Dental degree, Dental Degree, Dentistry degree, MD, Medical degree, Medical Degree	5

Table 2: Numerical value assigned to each (categorical) value of the education attribute

Model	Male	Female
`Llama 3 8B-Instruct`	72	55
`Ministral-8B-Instruct`	79	79

Table 3: Number of male and female individuals generated by each model

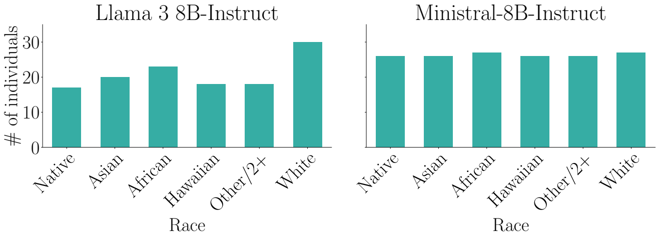


Figure 9: **Factual distributions of race.** The panels show the empirical factual distributions of race for the individuals generated by `Llama 3 8B-Instruct` (left) and `Ministral-8B-Instruct` (right).

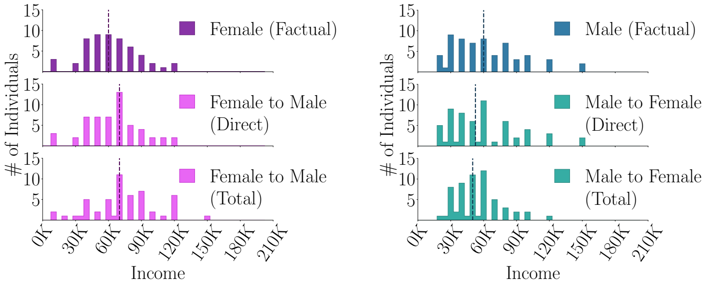
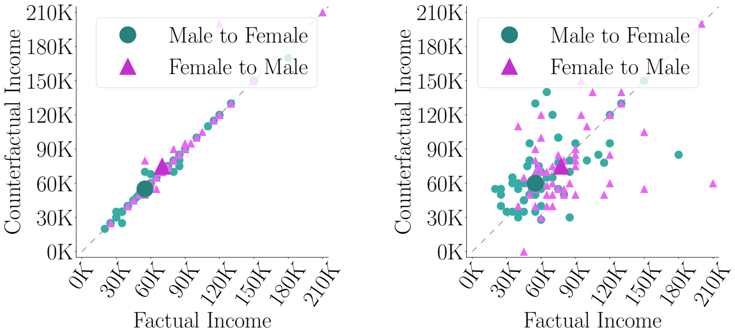


Figure 10: **Comparison between factual and counterfactual income.** The left (right) panel shows the factual distributions of income of female (male) individuals and their counterfactual distribution of income under two interventions: (i) had they been male (female), while keeping fixed the rest of their attributes preceding income in the output sequence, and (ii) had they been male (female), while keeping fixed the attributes preceding sex but allowing the attributes between sex and income to change in the output sequence. In both panels, dashed lines correspond to the median income. Here, we use `Llama 3 8B-Instruct` and set the temperature parameter to $\tau = 0.8$.

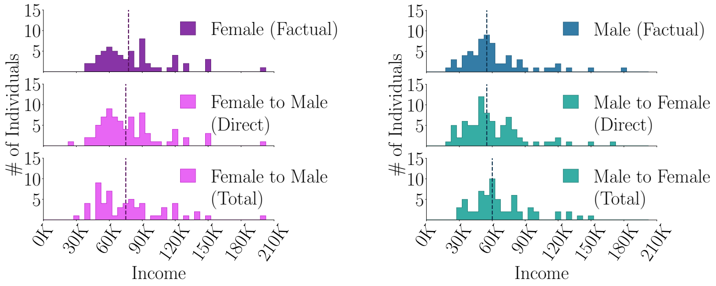
Appendix C. Additional experimental results on bias detection

Here, we present experimental results complementing those in Section 4.3, regarding the direct and total effects of sex on income and the total effects of race on education and occupation. Figure 10 complements Figure 5(*c*); it shows the distributions of the factual and counterfactual incomes of the individuals generated by `Llama 3 8B-Instruct`, under all possible interventions on their sex. Figure 11 complements Figure 5; it summarizes the results regarding direct and total effects of sex on income for the individuals generated by `Ministral-8B-Instruct`. Figure 12 complements Figure 6; it summarizes the results regarding total effects of race on education and occupation for the individuals generated by `Llama 3 8B-Instruct`.



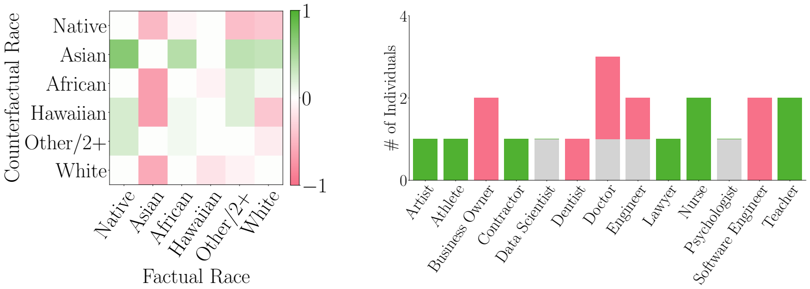
(*a*) Change in income upon intervention on sex (direct effect)

(*b*) Change in income upon intervention on sex (total effect)



(*c*) Distributions of factual and counterfactual income

Figure 11: **Comparison between factual and counterfactual income.** Panel (a) shows the change in income of male (female) individuals had they been female (male), while keeping fixed the rest of their attributes preceding income in the output sequence. Panel (b) shows the change in income of male (female) individuals had they been female (male), while keeping fixed the attributes preceding sex but allowing the attributes between sex and income to change in the output sequence. Panel (c) shows the factual distributions of income of female and male individuals and their counterfactual distributions of income under the same interventions as in panels (a) and (b). Enlarged points in panels (a, b) and dashed lines in panel (c) correspond to the median income. In all panels, we use `Ministral-8B-Instruct` and set the temperature parameter to $\tau = 0.8$.



(*a*) Change in education level upon intervention on race

(*b*) Change in occupation upon intervention on race

Figure 12: **Comparison between factual and counterfactual education and occupation.** Panel (a) shows the average difference in the education level of individuals of each race had their race been different. Here, positive values indicate an improvement in education and negative values indicate a decline. Panel (b) shows the distribution shift of occupations among Asian American individuals had they been Black or African American. Green (red) sections indicate the counterfactual increase (decrease) in the number of individuals that practice each occupation. In both panels, we use `Llama 3 8B-Instruct` and set the temperature parameter to $\tau = 0.8$.

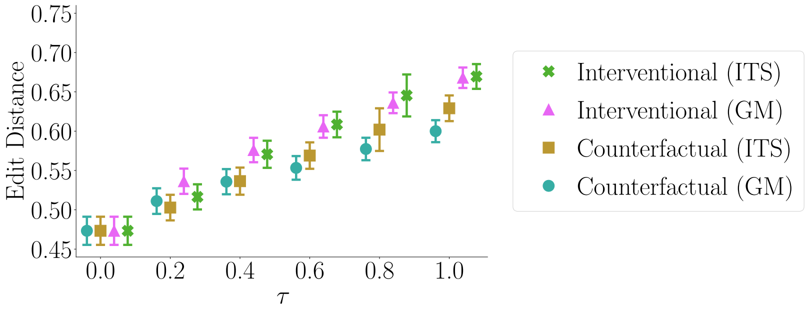


Figure 13: **Comparison between interventional and counterfactual token generation under two different SCMs.** The plot shows the edit distance between the factual sequence of tokens and the sequence generated by interventional and counterfactual token generation using the Gumbel-Max SCM (GM) defined by Eq. 3 and the SCM based on inverse transform sampling (ITS) defined by Eq. 4, against various values of the temperature parameter τ. The edit distance is averaged over 4,000 output sequences, resulting from two independent interventions per factual sequence, and error bars represent 95% confidence intervals. In this panel, we use `Llama 3 8B-Instruct`.

Appendix D. Additional experiments using a sampler that does not satisfy counterfactual stability

In this section, we conduct counterfactual token generation using a classical sampler for categorical distributions, which can be viewed as an SCM that is not guaranteed to satisfy counterfactual stability.

Experimental setup. We experiment with two different SCMs to implement the function f_T responsible for sampling the next token on `Llama 3 8B-Instruct`: (i) the Gumbel-Max SCM defined in Eq. 3 and (ii) an SCM based on inverse transform sampling defined as follows:

$$f_T(D_i, U_i) = \underset{j \in \{1,2,\ldots,|V|\}}{\operatorname{argmin}} \left\{ j \cdot \mathbb{1}\left[\sum_{k=1}^{j} D_{i,k} \geq U_i\right]\right\}. \tag{4}$$

Under the SCM based on inverse transform sampling, the next token is sampled as follows. Let the tokens $t \in V$ follow a fixed order across time steps i. To generate a token T_i, for each position $j \in \{1, 2, \ldots, |V|\}$, the SCM based on inverse transform sampling computes the cumulative sum of probabilities in the distribution D_i corresponding to the tokens in positions $k \leq j$. Then, it samples a unidimensional noise variable $U_i \sim \text{Uniform}(0, 1)$ and selects the first token in the vocabulary V whose corresponding cumulative sum is greater than or equal to U_i.

Similarly as in Section 4.2, we compare the edit distance between factual output sequences and sequences generated through interventional and counterfactual token generation using the Gumbel-Max SCM and the SCM given by Eq. 4.

Results. Figure 13 summarizes the results, which show that, under both SCMs, the output sequences generated using counterfactual token generation remain more similar to their respective factual sequence (*i.e.*, exhibit lower edit distance) compared to the sequences generated using interventional token generation. Moreover, as expected, the counterfactual output sequences generated using the SCM defined by Eq. 4, which does not satisfy the property of counterfactual stability, exhibit an edit distance higher than those generated using the Gumbel-Max SCM.