Defining and finding the gender direction in gpt2-xl

Léo Dana

October 19, 2023

This document is the internship report of my work done at FAR AI and SERI MATS in Berkeley under the mentorship of Claudia Shi during the summer of 2023.

I present my work to find a direction associated with gender in the language model GPT2-xl, with the goal of being able to control its gender biases. With such a direction, one should be able to predict the gender, erase gendered information, and change the gender inside sentences. My work draws inspiration from the literature on word embedding models [2], as well as new techniques applicable to LLMs [1, 23, 13]. I was able to design and conduct preliminary experiments that tested the feasibility of a real-time intervention for gender debiasing. Despite, some underwhelming results, I think it is yet too soon to conclude on the viability of the general method. The code of my experiments can be found here.

I would like to thank Claudia for mentoring me throughout those months and teaching me very helpful research advice, especially in computer science-like research which was new to me. I also thank Adam Gleave, CEO of FAR AI, as well as the FAR AI team and the SERI MATS team for hosting me, it was a great pleasure working at their lab and meet all these amazing people. Finally, I thank EffiSciences for helping me find this internship in the first place.

Contents

1 Introduction 3

1.1 AI risks 4

1.2 Addressing the risks 5

2 Language Model's architecture 6

2.1 The MLP block 7

2.2 The Attention block 7

2.3 Training objective 8

3 The Gender Direction 10

3.1 Causal intervention 11

3.2 Concept erasure 12

3.2.1 Quantile Leace 14

4 Experiments 17

4.1 Datasets 18

4.2 Gender classification 19

4.3 Steering and Erasing 20

4.3.1 Intervention at each layer 21

4.3.2 Intervention at all layers 23

4.3.3 Text generation 24

5 Conclusion 25

A Data Tables 29

B Additional graphs 31

1 Introduction

Deep learning models are state-of-the-art in many domains requiring general capabilities like image recognition, image generation, and natural language processing. The Transformer architecture [25], initially developed for natural language processing, is an example of a recent milestone that is now used in other domains like Reinforcement Learning with success.

However, this deep learning architecture is currently not understood insofar as we don't know how they generalize to new unobserved data. As a comparison, linear regression is more understood as we know which data it should generalize to, and how the answer is computed, once the algorithm is learned. In this case, each coefficient gives the extent to which each coordinate of the input is useful. For deep learning models, we do not have such interpretable circuitry.

This challenge is key to many real problems using AIs like self-driving cars: not knowing what the car is capable of doing puts the user at risk when the car is deployed in new environments. The same process happens with language models that have been jailbroken[1] into having aggressive behaviors.

One way to increase our understanding of such models is to perform causal mechanistic interpretability, which is a research field gathering the techniques that aim at finding the causal circuitry of the models and modify it using causal intervention [19]. This field wants to answer questions such as "How do models perform translation?", "How do models identify dogs on images?" or "How do models understand and compute gender?". The latter question is the main question this report tries to answer.

One particularly important type of deep learning model that should be made more interpretable is Language Models because of its widespread use. AIs like GPT4, Bard, or Claude will shape our society, so it is crucial that we understand them and make sure they are safe to use. This is not currently the case, and if interpretability for image classification seems on a good track [18], it is not the case for language models as they showcase very broad and complex capabilities. Researchers seem to have found new and unexpected circuits as they dive deeper into language models (see backup heads and negative heads [26]).

The biggest circuit understood so far in a language model is the *Indirect Object Identification* (IOI) [26] circuit[2]. But what defines a circuit is even not well-defined yet, as there is a tradeoff between the complexity of a circuit and its predictive power, as shown in [6]. This showcases that today, we cannot claim confidently AIs to be safe.

This is the reason why I was and will be working on language models to

[1] A jailbreak is a way to prompt a language model into saying things it was trained not to say. The most impressive of them can be found on twitter, jailbreaking the Bing AI.

[2] It answers the question: how can a model infer, in a sentence containing two characters, who is the one that is going to be referred to? "Mary and John like to play football. John often passes the ball to" - "Mary". In this example, IOI understands the circuit that enables GPT2 to know that John passes the ball to Mary, and not to itself.

make them more interpretable. In particular, I will be focusing on conceptual directions: find a direction in the LLM that represents a concept, such as gender. Gender will be the main example I use in this report because there is an important literature already focusing on gender equity in statistical models. Still, this work could be extended to finding concepts like truth, as in [4], or deception, as in [21], and increase AI's safety.

1.1 AI risks

In this subsection, I would like to give more context about what are the risks that we face with future AI systems in more generality.

The CAIS produced a list of 8 risks that could arise from general AI systems. 4 of these risks could be mitigated by doing technical alignment research. In this subsection, I will explain more about the risks, and in the next subsection I will detail possible solutions that have been proposed to mitigate them.

Goal Misgeneralisation: from the article [12], the authors show that it is not trivial to specify a goal in a way that makes your AI robustly pursue it. For example, their agent trained to grab a coin may only learn to go to the right side of the screen, because the objective was underspecified. It might be simple to correct in this example, but for real-life applications like autonomous vehicles, you don't want to plan in advance for every case, we want the AI to generalize the right way from the start.

Outer Alignment: If we had a simple goal to give to our AI systems, like maximizing points, we could focus our efforts on *Goal Misgeneralisation*. But what we want AI systems to do is often under-specified in the language, and might cause the wrong generalization of goals. It is philosophically hard to say what we mean by "not harm people", and don't have the same trade-off between society.

Power Seeking Behaviors: If an AI has for goal to maximize a quantity, some ways of achieving this might be by obtaining power and money to then pursue the maximization easily. These goals, acquiring power and money for example, are called instrumental in that they are not the goal-target, but are achieved as a way to maximize the target-goal. [24] shows that these Instrumental Goals are optimal for any agents that try to maximize their goals and have uncertainties.

Deception: This risk is speculative[3] as it has not been showcased anywhere, but is theorized to happen with very intelligent systems[4]. The idea is that if an agent has an instrumental goal, it could act to make us believe it is not a danger for us, but would pursue its other goal once we deploy it.

[3] This is however a serious research direction, pursued by lab like Anthropic

[4] By intelligent, I mean exhibiting impressive cognitive capabilities like GPT4, and pursuing goals.

1.2 Addressing the risks

The risks mentioned above are being tackled by many lines of work, none of which seems to fully solve the general problem. I detail below the main current ways to reduce the risks.

Policy work: The development of AI is fast and policymakers are lagging behind. For this reason, several actors in the AI field have requested a giant AI pause to give time for regulations and to give the public the details of how AIs are going to impact the future.

Interpretability: Most of the technical research in AI safety today is oriented toward interpreting models. This means finding circuits [6] that produce certain behavior to intervene in them and stop harmful or deceptive behaviors. As we said before, this line of research is very large and much more developed in vision models than in language models. Parallel to this field, other researchers try to design AI to be interpretable or explainable by design.

Evaluation: With AIs like GPT4, it is important to assess the capabilities of language models to know if they are safe to use. The most common evaluation concerns biases [2], but other evals have been targeting the potential of GPT4 to hack websites, solve captcha [3], answer moral dilemmas [22], and be situationally aware.

Other works : New research directions have started to arise that are preparadigmatic like classifying language models' "mental state" [17, 11], automating AI safety research with AIs with superalignment, etc.

2 Language Model's architecture

In this section, I will introduce the main AI model that I will use throughout the report to experiment with.

I worked on the Large Language Model **GPT2-xl** [20] which is a 1.5 billion parameters Transformer. The model takes as input any sentence, decomposes it into tokens, and outputs a probability distribution representing which token is most likely to be seen next. With this simple goal of next token prediction, the model learns quite general capabilities, as showcased by OpenAI's GPT4, Google's Bard, or Anthropic's Claude.

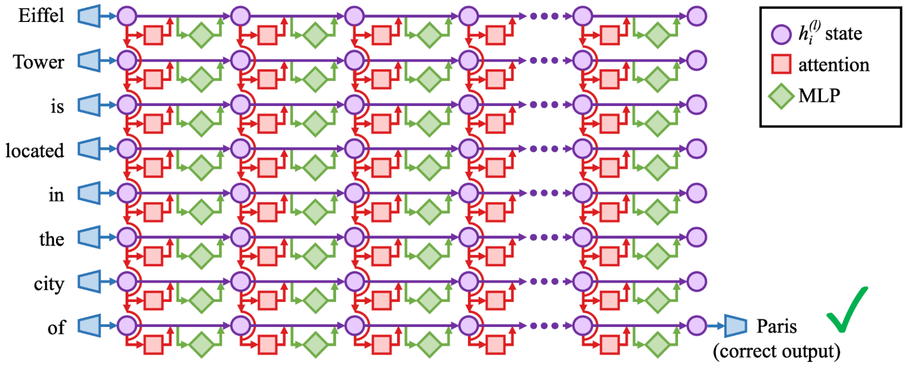

Figure 1: Transformer architecture: each token of the sentence is computed in parallel, but only the last one is used to make the prediction. Importantly, the set of parameters used to compute each token is the same. Image from [14].

Gpt2-xl is a decoder-only Transformer, and works as follows:

- Any sentence is first converted into a set of tokens (there are 50256 different tokens). The *tokenization* is performed by another function and is very unintuitive.

- Each token is then embedded into a 1600-dimensional vector using a *word embedding* and a *positional embedding.* The word embedding is simply a table associating each token to a vector. The positional embedding is a vector that represents the position, or stream of the token in the sentence.

- They are then added to the *residual stream.* It is an abstraction used to denote the place in the network where the computed vectors are placed (the dark blue lines in Figure 1). Vectors from the residual are called *activations.*

- This vector is then passed on to several layers of computations. Every layer is composed of an *Attention block* (red squares) and an *MLP block* (green diamond). Each component contains a *skip-connection* meaning that the output of the component is just added back to the residual stream.

- The Attention block makes each residual stream communicate, it is the key component of the success of this architecture.
- After going through each layer, each vector is unembedded into a distribution over tokens.
- It represents the probability that the next token is seen after the input sentence, which is the objective that most Transformers are trained to optimize.

I will now explain in detail how the mlp and attention block work, and what the training objective of language models is.

2.1 The MLP block

The *Multi-Layer Perceptron* (MLP) block is the standard non-linear dense layer used in Neural Networks. It doesn't interact with other streams, so we can express easily what it does on the residual stream. Let $x \in \mathbb{R}^{1600}$, we have

$$MLP(x) = W_2 \sigma_G(W_1 LN_{\alpha,\beta}(x) + b_1) + b_2$$

where $\sigma_G(y) = \text{GELU}(y) = y\mathbb{P}(Y \leq y)$ with $Y \sim \mathcal{N}(0,1)$, and $LN_{\alpha,\beta}$ is the Layer-Norm function. The GELU activation functions designed to be a smooth approximation of the standard ReLU activation. The layer norm rescales the vector x before passing it through the MLP with

$$LN_{\alpha,\beta}(x) = \frac{x - \frac{1}{n}\langle x|\mathbb{1}\rangle}{||x - \frac{1}{n}\langle x|\mathbb{1}\rangle||_2} \cdot \alpha + \beta$$

where $\mathbb{1} = (1, ..., 1)$, α and β are learned parameters, and $\cdot$ denotes the component-wise multiplication. Finally, $W_1 \in \mathbb{R}^{1600,6400}$ and W_2 are matrices, and b_1, b_2 are biases.
The output of the MLP is then added back to the residual stream using a skip connection, so the hole operation is $x \to x + MLP(x)$.

The MLP mechanism remains today the least understood component of the architecture, despite attempts to link it to memory storage [14, 15].

2.2 The Attention block

The *Self Multi-Head Attention Layer* was introduced in [25] and is the main component that gives its specificity to the Transformer architecture because it allows each token's stream to communicate. We will detail how it works step-by-step. Let $i \in [1, N]$ the token stream, and $x_i \in \mathbb{R}^{1600}$ the vector of the residual at this stream and layer. We choose to not denote the layer number because every operation happens at the same layer, even if each matrix and layer-norm has different parameters depending on the layer.

1. We start by computing the Layer Norm $X_i = LN_{\alpha,\beta}(x_i)$,
2. Then, compute linearly three new vectors

$$\text{key}_i = W_{key}X_i,\ \text{query}_i = W_{query}X_i,\ \text{value}_i = W_{value}X_i,$$

3. Now we divide each vector into subcomponents that are going to go in different *Attention Heads*. In GPT2-xl there are 16 heads per attention, each of dimension 100. Thus, we now denote each vector with its head number h,
4. For stream i, we then compute the coefficients $\mu^h_{i,j} = \langle \text{query}^h_i | \text{key}^h_j \rangle$ for every $j \leq i$ and then take $\{\nu^h_{i,j}\}_j = \text{softmax}(\{\mu^h_{i,j}\}_j)$, with

$$\text{softmax}(\{a_j\}) = \left\{ \frac{e^{a_j}}{\sum_k e^{a_k}} \right\}_j$$

5. We compute the final value per head

$$Value^h_i = \sum_{j=0}^{i} \nu^h_{i,j} value^h_j,$$

and then concatenate back each head's value to output $Value_i$,

6. So the computation of the whole block is $x_i \rightarrow x_i + Value^0_i \bullet ... \bullet Value^{15}_i$.

The attention block allows a stream to query information from previous streams that have a matching key for the demand. It is also almost completely linear: a non-linear reweighting of the linearly queried streams.

Some work has already been done on interpreting how Attention works formally [9] and finding circuits in GPT2 that perform meaningful computation [26].

2.3 Training objective

Transformers like gpt2-xl are trained using the task of next token prediction, meaning that given a sequence of tokens [’Je’, ’ m”, ’appelle’], they have to predict the next token of the sentence T = ’ leo.’. The model will predict a probability distribution p over the next token, and this distribution is compared to the Dirac distribution δ_T using the cross-entropy loss:

$$\text{CrossEntropy(p,q)} = -\sum_i p_i \log(q_i)$$

This pretraining phase uses up to 500 billion data points, which is even not enough for the number of parameters according to DeepMind's scaling laws [10].

Nowadays, the training of these huge language models is often followed by a Reinforcement Learning via Human Feedback phase [5] to make the model easily usable and harmless. First, an evaluation model is trained to compare sentences and tell which one is the best, as defined by human evaluators. Then, This smaller model is used in place of human evaluators to train the model to generate sentences with a higher score. Once the model is tuned using RLHF, it becomes way better at answering questions that we ask it, and worse at using offensive language, etc.

3 The Gender Direction

The goal of my project is to find a direction for the gender concept to steer, erase and predict gender in a model, both on questions and text generations. Let's see what this means.

By direction, I mean a unit vector $d_l \in \mathbb{R}^{1600}$ of the residual stream at layer l, before the attention block l, as well as a bias b_l, so we learn an affine hyperplane. We will have one direction per layer because there is no guarantee that the direction will not change through the layers. This vector will represent the gender direction, so is related to the distribution of residual stream activations $X_l \sim \mathcal{D}_l$ and its gender $Z_l = \text{gender}(X_l)$, and it should have 3 properties:

- Erasing gender: if X_l is an activation of a token at layer l, then $\tilde{X}_l = X_l - \langle d_l | X_l - b_l \rangle d_l$ is the activation without gender. So if the model had understood X_l as produced by a male token, then $\tilde{X}_l$ should confuse the model on which gender it represents:

$$\mathbb{P}(Z_l = \text{male}|\tilde{X}_l) = \mathbb{P}(Z_l = \text{female}|\tilde{X}_l)$$

- Steering[5] gender: in the same context, on top of erasing gender, we could be able to change the gender in the direction we want by doing $\bar{X}_l^\lambda = X_l + \lambda \langle d_l | X_l - b_l \rangle d_l$:

$$\mathbb{P}(Z_l^\lambda = \text{male}|\bar{X}_l^\lambda) > \mathbb{P}(Z_l^\lambda = \text{female}|\bar{X}_l^\lambda) \iff \lambda > 1$$

  Note that when $\lambda = 1$, we are doing gender erasure.

- Predicting gender: we use the predictor $\text{sign}(\langle d_l | X_l - b_l \rangle)$ to distinguish male and female activation. It should have a "good" accuracy (significantly better than random):

$$Z_l = \text{sign}(\langle d_l | X_l - b_l \rangle)$$

Finally, these properties should not just hold when evaluated on a dataset of question-answers, but also in a text-generation setting[6].

The project as a whole is quite ambitious, and the hope is to make significant progress on knowing if it is possible to find such a direction, and if not which hypothesis would not hold: maybe there is not only one direction, maybe steering and predicting are not the same direction, maybe the abstraction of a direction is not the right one, and we should instead be looking for circuits in the network (as in [26]).

[5] The basis of steering activation vector was introduced in [23].

[6] A classical failure mode of interpretability is that good performance on benchmarks can hide that the method is too destructive, and the model is no longer able to generate meaningful prompts.

In the rest of the section, I present the two main tools that I will use to find the gender direction. These are *causal intervention*, which lets you analyze the counterfactual behavior of the model in different settings, and *concept erasure*, which allows you to suppress information linearly from the model's activation.

3.1 Causal intervention

Let $M = (V, E)$ be an acyclic-directed graph. M can represent a *Structural Causal Model* as described in [19] where the vertexes are some random variables, and the edges are causal relationships between those variables. The formalism of Causality can help us design tests to study how two variables are linked in a model, knowing only the structure of the model. In general, the causal model is not known but is the researcher's hypothesis. Let me illustrate this with an example:

$$Z = X \bmod[Y],\ Y = \min(\lfloor \log(X) \rfloor, 10),\ X \in \mathbb{N}^*.$$

$$\begin{tikzcd}
X \arrow[rr] \arrow[dr] & & Z \\
& Y \arrow[ur] &
\end{tikzcd}$$

Here, if we want to understand the function by only looking at the variables X and Z, it will be quite complicated to guess that we have

$$Z = X \bmod[\min(\lfloor \log(X) \rfloor, 10)].$$

Using the variable Y makes the job easier, and for that, we need to perform some *causal intervention*. For example, if we wish to understand the effect of X on Z, we need to look at the direct path $X \to Z$, and the indirect path $X \to Y \to Z$. To study only the direct one, we can decide to intervene on Y by setting the variable $Y = y$ to a fixed value and looking at how Z changes when X varies. In practice, we can equivalently filter all experiments to only keep the ones with $Y = y$. So, by fixing $Y = 10$ and modifying X, we can realize that $Z = X \bmod[10]$, and by varying Y, we understand that $Z = X \bmod[Y]$.

Causal intervention is a very powerful method to understand the effect of a variable on another in a complicated causal model. This technique is then a good standard for studying LLMs' internal representation. Indeed, we are in the same setting where the function $Z = f(X)$ is too complicated to understand, so we need to introduce intermediate variables $\{Y_k\}$ to ease the work. However, one needs to be very careful in continuous settings about out-of-distribution intervention: if I intervene on Y by fixing it to a value it cannot take, the interpretation that I have of the algorithm will be incorrect. For example, if I perform the intervention $Y = 15$ on the previous model, then I might believe $Z = X \bmod[\lfloor \log(X) \rfloor]$, and miss the $\min(\lfloor \log(X) \rfloor, 10)$.

The same phenomenon can happen inside neural networks: by replacing the input of an Attention block with another value, I might put the whole network out-of-distribution, *e.g.* the computation that the network will do will not be representative of what the network usually does, and we can falsely interpret the network this way.

In the literature, there exists evidence that causal intervention making the model out-of-distribution worsens the analysis. For example, [26] operates causal intervention to find a circuit in GPT2 doing indirect object identification. For that purpose, they use a method called *ablation*: to see the effect of an attention head on the answer, they performed a causal intervention by setting the output of the head to 0. However, since the vector 0 is very unlikely to appear as the output of this head, this puts the network out-of-distribution and leads to noisy interpretations. They found that replacing the output of the head with the mean of the usual output of this head yields better results.

Together, the articles [23, 26] make the case that one should use the activations that the network produces in order to perform good causal intervention[7]. This influenced me in my work in two ways:

1. When constructing the gender direction, I will use gpt2's activations directly instead of learning via an optimization process[8]. This property is satisfied by the Leace technique I will use, and present in the next section.

2. There is always the possibility that I will be learning a direction that makes my network behave in weird ways when steered or erased, thus I need to evaluate the technique both on a well-defined question-answer dataset, but also on text generation. This means creating sentences using the model and measuring how much the behavior is altered by the erasure or steering. The latter criterion is harder to measure in general but is closer to the target goal of modifying SOTA models.

3.2 Concept erasure

Concept erasure refers to the idea of being able to "suppress a concept or information from data", like gender. Formally, for random variables X, Z, we want to construct $f(X)$ such that it is impossible to predict Z using $f(X)$ better than a constant predictor. We say in this case that $f(X)$ *guards* Z. In the case where Z is a binary variable representing gender, this is equivalent to creating a non-gendered representation of X. This problem is intractable in full generality. If $Z = X$, then we need to take $f(X)$ constant, which is the trivial solution and is useless in practice: we suppress all information from X, so we can't predict Z, but also can't use X anymore.

To overcome this problem, one needs to restrict the space of classifiers that one can use to predict Z from X. The *Leace* technique [1] gives a closed-form

[7] One should try to speak "the language of the network", as an analogy to interpreting biological systems.

[8] In general, it seems that optimization processes are less robust because they might abuse some pathways in the network that are very out-of-distribution. One of my earlier projects involved such ideas, and I suspect that did not work for that reason.

solution in the setting where we restrict ourselves only to linear predictors. This enables us to then perform linear concept erasure[9].

Let $X \in \mathbb{R}^n$, $Z \in \{0,1\}^k$ such that $\sum_i Z_i = 1$, $\mathcal{L}$ a set of convex losses, and $\eta_\theta : \mathbb{R}^n \to \mathbb{R}^k$ a family of predictors parametrized by Θ. In the case $k = 2$, this is equivalent to having $Z \in \{\pm 1\}$.

Definition 1 (**Guardedness**). *We say that $X - (\Theta, \mathcal{L})$ guards Z if for any $L \in \mathcal{L}$,*

$$X \in \underset{X'}{argmax} \underset{\theta \in \Theta}{inf} \mathbb{E}\left[L(\eta_\theta(X'), Z)\right]$$

This says that the distribution X is the best to predict Z on any predictors and any losses. We say that X linearly guards Z if $\mathcal{L}$ is any convex loss function, and $\eta_{P,b} = PX + b$ is any linear transformation.

Definition 2 (**Trivially attainable loss**). *We call $l_L = \inf_{b \in \mathbb{R}^k} \mathbb{E}[L(b, Z)]$ the trivially attainable loss, which is the best loss obtained by a constant predictor.*

We can now restate properly the linear guardedness result from [1]. For simplicity, we only gave the equivalent formulations that are useful for the report.

Theorem 1. *We have the following equivalences.*

1. *X linearly guards Z,*
2. *For any convex losses L, $\sup_{P,b} \mathbb{E}[L(PX + b, Z)]$ is greater than the trivially attainable loss,*
3. *$\Sigma_{XZ} = Cov(X, Z) = 0$*

This theorem means that for X to linearly guard Z, it is equivalent that no linear predictor has a better loss than the trivial loss, and is equivalent to having uncorrelatedness between X and Z. This is really powerful as it translates an optimization problem into an equality to satisfy.

Going back to our original problem, we want to find f such that $\mathrm{Cov}(f(X), Z) = 0$. One convenient choice of the family from which to choose f is linear functions. The intuition here is that we want to remove some directions from X to guard the concept of gender. Transforming linearly X will enable us to remove the direction while changing X as little as possible[10].

[9] One could wonder why linear guardedness would be sufficient if Neural Networks are mostly non-linear. There is evidence that even if the model's way of doing computation is non-linear, the way the information is represented is in some cases linear [16].

[10] This is particularly important because we want the causal intervention to be in-distribution as much as possible.

This gives the equation $\text{Cov}(PX + b, Z) = P\Sigma_{XZ} = 0$ which is equivalent to having $Im(\Sigma_{XZ}) \subset Ker(P)$. This constraint still gives a lot of possible choices for P, so we choose the one that has the minimal L^2 effect on X.

Theorem 2. *The optimization problem*

$$\inf_{P\Sigma_{XZ}=0} \mathbb{E}[||PX + b - X||_2]$$

is satisfied by $P^* = Id - W^+(W\Sigma_{XZ})(W\Sigma_{XZ})^+W$ *with* $W = \left(\Sigma_{XX}^{1/2}\right)^+$ *the whitening matrix,* $^+$ *is the pseudo-inverse operation, and* $b^* = \mathbb{E}[X] - P\mathbb{E}[X]$.

We can interpret this solution geometrically. First, the matrix W is the empirical whitening matrix of the distribution $\mathcal{D}_X$. This mean that $\text{Var}(WX) = Id$ when Σ_{XX} is invertible. In this case, we can rewrite $P^* = W^+(Id - (W\Sigma_{XZ})(W\Sigma_{XZ})^+)W$. We can now see more clearly what is the effect of P^*. We first shift the distribution of X to make it standard, then we take the orthogonal projection onto $W\Sigma_{XZ}$ which is the difference in mean between the different classes when the distribution is standard, and finally, we scale the distribution back to its normal variance. This solution is better than the naive projection $Id - \Sigma_{XZ}\Sigma_{XZ}^+$ because it takes into account the internal covariations between coordinates of X, as shown in Figure 2.

3.2.1 Quantile Leace

An important feature of *Leace* is that no linear predictor can perform better than a constant predictor. This is a claim about the accuracy of linear models, but we would like our model to also have an equal probability of outputting each gender:

$$\mathbb{P}(Z_l = \text{male}|\tilde{X}_l) = \mathbb{P}(Z_l = \text{female}|\tilde{X}_l).$$

I will compare here the choice of the parameter b^* when the model is a linear predictor, and when the model has a threshold. This work is my contribution to the previous work [1] of Nora Belrose with whom I was able to discuss.

Linear model: Let $Z \in \mathbb{R}$, and imagine that $Z = \delta \cdot (X - c)$, then $\delta \cdot P^*X = 0$, and at inference time after erasure, we have $Z = \delta \cdot (b^* - c) = \mathbb{E}[Z]$. Thus, the predictor can be biased toward one of the classes if the labels are not carefully balanced. One can mitigate this effect by taking $\bar{b} = b^* - \mathbb{E}[Z]\frac{\delta}{||\delta||_2}$. Therefore, we can trade off preserving the L^2 distance from X, with having an erasure that ensures that no class is preferred at prediction time.

Threshold model: The above model is too simple for Neural Networks because the next token prediction has a threshold (we take the token that has the greatest probability). In such cases, the above modification of b^* is not

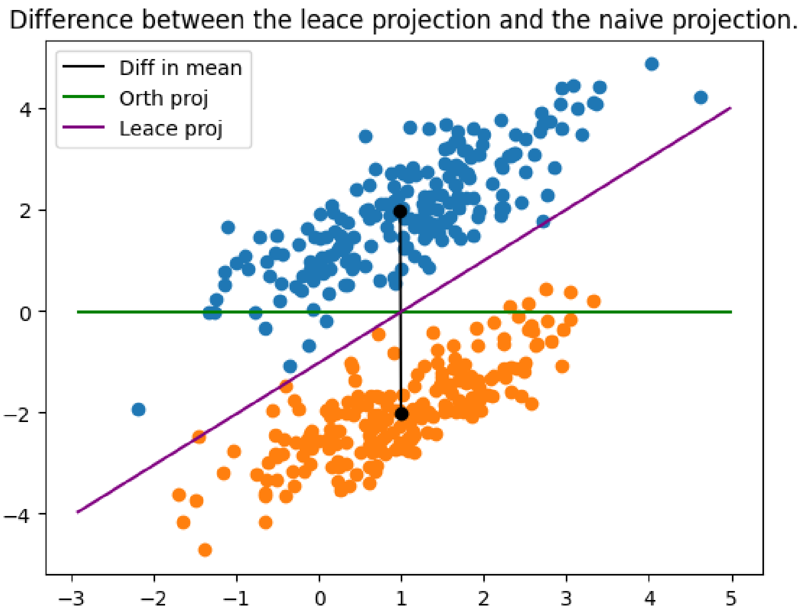

Figure 2: Synthetic Gaussian distributions with non-axis aligned covariance. In black is the difference in mean, which is also in this case the direction Σ_{XZ}. The naive projection space (green) doesn't separate the distributions nicely, but the leace projection space (purple) does.

sufficient, see Figure 3. We can model that $Z = \mathbb{1}_{\delta\cdot(X-c)\geq 0} - \mathbb{1}_{\delta\cdot(X-c)\leq 0}$. The predictor is no longer linear, so we don't have the guarantee that it is worse than a constant predictor, and in general it is not. By using concept erasure here, we still have a constant predictor, and to make it class agnostic, we want that $b^* = c$ as before (we project on the decision boundary). Moreover, we have $\mathbb{E}[Z] = 2\mathbb{P}(\delta \cdot (X-c) \geq 0) - 1 - \mathbb{P}(\delta \cdot (X-c) = 0)$. We can approximate this probability by $\frac{1}{n}\sum_i \mathbb{1}_{\delta\cdot(X_i-c)\geq 0}$. So, if X has no atoms, we have

$$\frac{1}{n}\sum_i \mathbb{1}_{\delta\cdot(X_i-c)\geq 0} = \frac{1+\mathbb{E}[Z]}{2}$$

which is approximately satisfied by $c = Q_{\frac{1+\mathbb{E}[Z]}{2}}(\delta \cdot X)\frac{\delta}{||\delta||_2}$, the $\frac{1+\mathbb{E}[Z]}{2}$ quantile of the distribution of $\delta \cdot X$ in the direction δ.

In the rest of the work, I will use this last modified version of *Leace* for concept erasure in my experiments. In particular, this means that the "conceptual

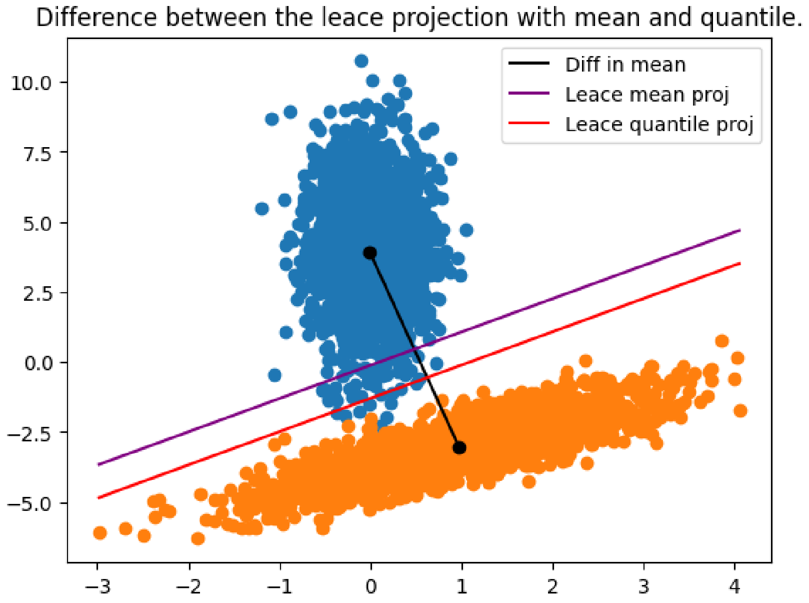

Figure 3: Synthetic Gaussian distributions with non-axis aligned covariance. Using the quantile projection (in red) leads to a better separating hyperplane than the mean projection (in purple), especially when the class covariance is not equal in the difference in means direction.

direction" is represented by two directions and a bias:

$$\begin{aligned} d_{\text{mean}} &= W^{+}W\Sigma_{XZ}\ (\approx \Sigma_{XZ}) \\ d_{\text{plane}} &= (W\Sigma_{XZ})^{+}W \\ b_{\text{quantile}} &= Q_{\frac{1+\mathbb{E}[Z]}{2}}(d_{\text{plane}} \cdot X)\frac{d_{\text{plane}}}{||d_{\text{plane}}||_2} \end{aligned} \tag{1}$$

with d_{mean} being almost the difference in mean (and exactly when W is invertible), (d_{plane}, b) being the affine hyperplane projection on which we project in the direction d_{mean}. I will perform my experiments with both b_{quantile} and b^{*} to have a point of comparison.

4 Experiments

In this section, I will describe my experiments and then analyze my results. The section is in part incomplete because my internship ended when the experiments were taking place. So some of them won't be in the report, but I still intend to finish them and write a workshop paper if possible. This section will only contain the analysis of the gendered words, and the analysis of the gendered names will be discussed in the appendix.

The experimental setup is composed of two parts: learning the directions using the Leace technique and evaluating them via causal intervention.

As stated in the last section, the primary hypothesis of this work is that there exists, at each layer, a direction that encodes the gender and that the network uses it to compute the next token. In general, we are interested in intervening on the network as little as possible while modifying the gender inside a sentence. For example, in the sentence "The woman has her/his leg broken. Pronoun:", the next token will be either "his" or "her"[11], so there are two different streams where to intervene:

1. The last stream: We change the gender in the stream that is directly computing the next token,
2. The "woman" stream: We change the gender in the stream where the word woman appears so that the rest of the network sees it differently,

To have the least intervention possible, one should intervene in the "woman" stream. This allows us to change only one token to intervene on all later tokens, so in the case where we would like to ask another question afterward, we don't have to make another change.

Following the same line of thought, this also suggests that one should modify only the part of the "woman" stream that is perceived by the tokens afterward. We thus propose a new intervention inspired from [1, 13]: to modify the cache after computation in the attention partner. So for s the indices of the stream we target, we perform the intervention below: after computing the attention pattern for the stream s, we intervene on the keys and values perceived by the following stream. This means that the stream s is computed as usual, but also that the streams afterward will not see the stream s as containing gender information. This is summarized in the diagram below.

[11] Using few-shot learning, it is possible to make the model answer only one of these two tokens.

$$\begin{tikzcd}
\text{query}^s \arrow[dr] & \{\text{key}^i\}_{i<s-1} \arrow[d] & \{\text{value}^i\}_{i<s-1} \arrow[dl] \\
\text{activation}_l^s \arrow[u] \arrow[r] & \text{Attention} \arrow[r] \arrow[d] \arrow[dr] & \text{activation}_{l+1}^s \\
 & \{\text{key}^i\}_{i<s} \arrow[d] & \{\text{value}^i\}_{i<s} \arrow[dl] \\
 & \text{Intervention} \arrow[d] \arrow[dr] & \\
\text{query}^{s+1} \arrow[dr] & \{\bar{\text{key}}^i\}_{i<s} \arrow[d] & \{\bar{\text{value}}^i\}_{i<s} \arrow[dl] \\
\text{activation}_l^{s+1} \arrow[u] \arrow[r] & \text{Attention} \arrow[r] & \text{activation}_{l+1}^{s+1}
\end{tikzcd}$$

To make this modification, one can simply learn the leace directions on the activations at each layer after taking the layer norm. So given a training dataset $\mathcal{D}$ of sentences containing gendered tokens, we have X_l the activation at layer l after the layer norm and at the stream which computes the gendered token. This relatively simple choice is because we want to modify the activation while staying as in distribution as possible. For Z, it will be 1 if male and -1 if the token is female.

4.1 Datasets

For the training dataset $\mathcal{D}$, I use the md_gender_bias dataset [7] hosted on Hugging Face. In particular, it contains data on the most popular names in the US between 1880 and 2018, as well as 222 pairs of the most gendered words in English. The activation addition literature ([23]) has shown that the use of pairs of activation ("he" - "she") is important to find a robust conceptual direction. Moreover, words and names were embedded in a sentence to make the language model compute more gender direction, and to average perturbation over many prompts. Examples can be found in the tables *Training Set* and *Training Prompts* of the appendix.

For gendered words, I separated them into three categories: nouns, pronouns, and anatomy. In each of them, I made sure they were ordered by pairs and suppressed words that were too ambiguous or uncommon.

For gendered names, I only choose to use the names from 1880 that had more than 20 people using them. This is because the data below 20 users was composed of names, the gender of which the model could not guess, which harmed performance.

For the test dataset, it is different depending on whether I will be testing the classification, or the erasing and steering. Examples can be found in the table *Testing Set* of the appendix.

In the case of the classification, I use the training dataset at 90% train - 10% validation to evaluate them. The separation was made on the words, so 10% of the words were not used to learn the direction, but they were all embedded in the same prompts.

Regarding erasing/steering, I asked ChatGPT-4 to create sentences containing one character denoted by one of the words in the training data. The task was to give the gender of the character, by choosing between 'male' or 'female'. I used few-shot learning[12] to make the model better at answering the question, but without making it extremely biased toward one answer, which is non-trivial when dealing with small models.

Finally, for the text generation, I did not have the time to properly create a dataset of sentences, so I will simply showcase some examples of generation success and failure.

4.2 Gender classification

For the prediction task, we use the predictive hyperplane $\text{sign}(\langle d_{\text{plane}} | X - b_{\text{quantile}} \rangle)$, where $d_{\text{mean}}, d_{\text{plane}}, b_{\text{quantile}}$ are the parameters learned using the leace method. We will compare the accuracy of that hyperplane with a logistic regression learned from the same data. This baseline will help to evaluate the quality of the dataset since a low accuracy with the logistic regression would mean that the data is not linearly separable.

Figure 4 shows a high accuracy of almost 1 for both the Logistic regression and the leace hyperplanes before layer 10, the accuracy drops afterward at 70% for the logistic regression and 60% for the leace hyperplanes. This accuracy stays the same on the train-validation dataset of Figure 5, except for the quantile hyperplane, which means that the directions learned are generalizing to unobserved data.

An interesting phenomenon to observe is the sharp drop in accuracy from layers 9 to 11. Before layer 9, the gendered words are well clustered, and after they seem to get mixed together, decreasing the accuracy. Finally, the accuracy increases back in the end, right before prediction. This could be explained by the superposition hypothesis: neural networks have the ability to perform computation in superposition, meaning that several concepts are stored in one dimension, or equivalently that one concept is stored in multiple dimensions, as proposed in [8].

Another hypothesis is that, even if the data is not separable by a hyperplane, the pairs of points are. Indeed, we are using pairs of points, and we

[12] Few-shot learning is a technique consisting of showing a few examples of the task the model should do, to make it more likely to do the task. This has been shown to drastically improve models' abilities.

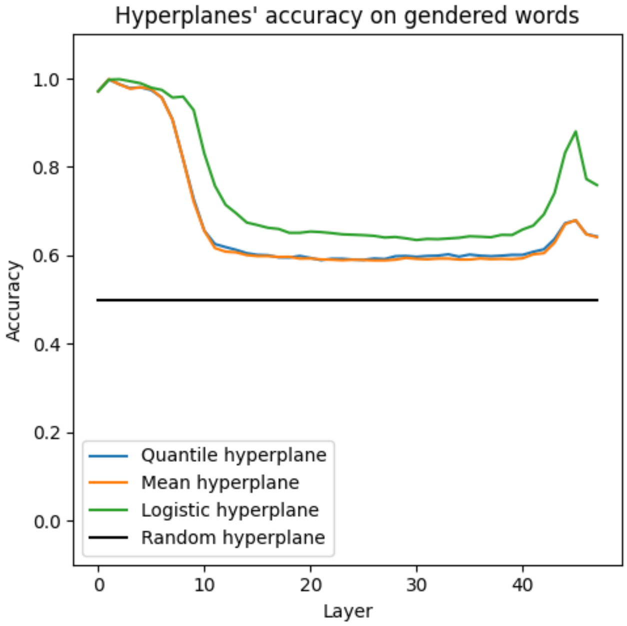

Figure 4: Accuracy of the Logistic regression, the quantile hyperplane, and the mean hyperplanes measured at each layer in the residual stream. We learned the hyperplanes on all the training data and tested them on the same data.

try to separate their activations, but we could try to separate the difference between their activations. Generally, this is simpler because we are just asking that $\text{sign}(\langle d_{\text{plane}} | X_{\text{male}} - X_{\text{female}} \rangle) > 0$, which removes the biases. This is indeed the case, and accuracy increases by 10%, but all pairs are not always in the same direction.

This result is very important for the upcoming experiments. Because it is easier to classify pairs of direction that points alone, it means that it will be more difficult to perform erasing than steering, since steering requires only knowing the direction to add, while erasing requires that the data is separable by a hyperplane.

4.3 Steering and Erasing

The evaluation for steering and erasing gender will be the same, but with different results expected. We will run 3 experiments:

1. We will measure the effect on the accuracy of the steering and erasing on each individual layer by steering and erasing in the residual stream.
2. Another experiment will be the erasing and steering applied at all layers

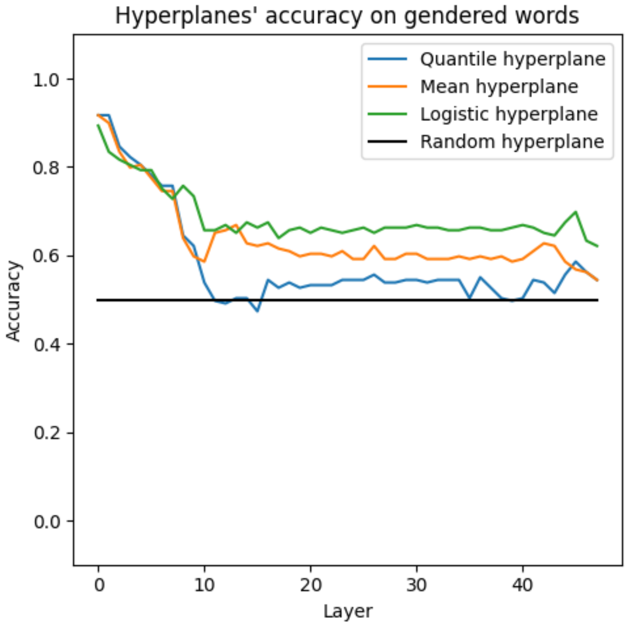

Figure 5: Accuracy of the logistic regression, the quantile, and the mean hyperplanes measured at each layer in the residual stream. We learned the hyperplanes on 90% of the training data and tested on 10%.

at the same time by targeting only the cache[13] of the attention pattern, and once again looking at the effect on the accuracy.

3. Finally, to ensure that the latter technique doesn't destroy the abilities of the language model, we will evaluate the model on the generation of a sentence. The steering and erasing methods should then modify the behavior of the model while keeping the sentence sound[14]. This last experiment is more qualitative since it is hard to assess automatically if a generation "makes sense". So following [23], we will present the best of 3 generations for several prompts, to show whether the technique works as expected or not.

4.3.1 Intervention at each layer

The results of this first experiment are displayed on Figure 6 for gendered words tested on female words. Graphs for gendered words on males and gendered

[13] The cache is the key, query, and values of the self-attention that are stored for later computations

[14] This last requirement is very important but also overlooked in the literature. Early experiments have shown that it is possible to satisfy the first test at the expense of making the model always output "she", which is detected by this last test.

names on females can be found in the appendix.

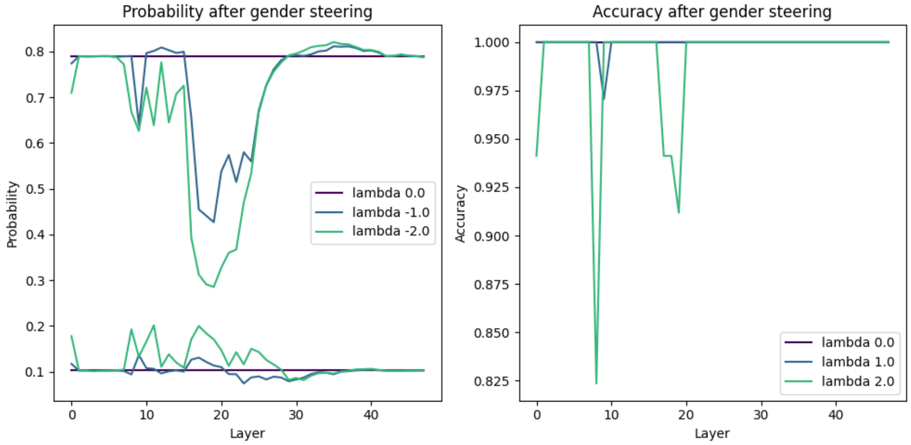

Figure 6: Probability and accuracy of intervention at each layer on female words. The lambdas represent different steering coefficients, with $\lambda = 0$, being the non-modified computation, and $\lambda = 1$ being the leace erasure.

Recall that the formula for gender steering is

$$\bar{X}^{\lambda} = X + \lambda \langle d_{\text{plane}} | X_l - b_{\text{quantile}} \rangle d_{\text{mean}},$$

thus, $\lambda = 0$ corresponds to the non modified model, and $\lambda = 1$ corresponds to performing erasure.

The figure shows a large effect on the output probability for both $\lambda = 1, 2$, but a small effect on the accuracy. In particular, it looks like the effect is dissymmetric on gender: the female tokens' probabilities are greatly decreasing while those of the male tokens are not increasing much. However, this might be a particularity of gendered words since Figure 10 on gendered names doesn't showcase the same behavior.

Moreover, all three figures support the evidence that the gender information is located or transmitted between layers 10 and 30. However, one needs to be careful with this interpretation: the task here shows where the information is perturbed most effectively, which is not directly a measure of where the information is. It would be possible that the intervention does something unexpected that makes the model bad at gender prediction, rather than actually erasing gender itself.

Together with Figure 10 and Figure 11, this graph comforts us with the idea that both erasing and steering are possible if we can interfere less with the standard functioning of the LLM to avoid destroying the computation, which happens for higher λ. This is the core idea of our next experiment.

4.3.2 Intervention at all layers

For this second experiment showcasing erasure and steering, we intervene on all layers at the same time, only on the cache of the attention mechanism as explained earlier. Figure 7 displays the results when the leace estimator is trained on gendered words, and tested on female questions.

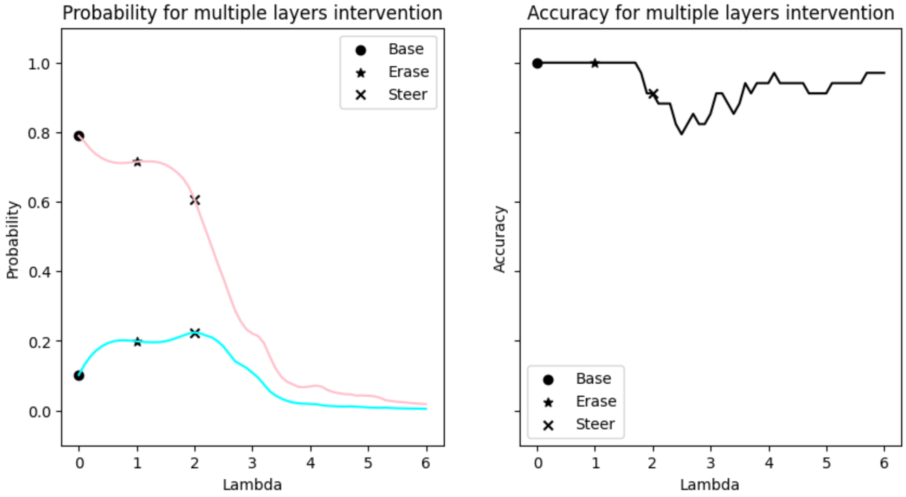

Figure 7: Probability and accuracy when intervening on all layers simultaneously, with leace learned on the gendered words and tested on the female words sentences. The axis Lambda is the amount of steering done at each layer, as explained in the previous section. $\lambda = 0$ is the non-modified computation, and $\lambda = 1$ correspond to the leace erasure.

First, we note that even if the erasure makes the probability of male and female closer to each other, meaning that the model is more confused about what's the correct answer, the accuracy stays the same. This also means that we likely won't be able to test the erasure and steering on text generation, as the most likely token will stay the same[15]. Another interesting finding, not present on the graph, is that even if the logistic regression separates the data in a better way, it doesn't erase gender well, proving that the leace direction is truly a better direction for this task.
Second, for the steering at high λ, we observe several intriguing facts:

- In both steering directions, the probability collapses once too much direction is added. This makes sense intuitively, we destroy the model activations, but goes against the findings in [23], even if the setting was slightly different.

[15] This could not be true, as the probability plotted is the average of the probability of gendered words.

- There is no amount of vector added that will change the predicted gender by a margin: at most, we can equalize the probability, but not make the curve cross.
- $\lambda = 2$ seems to be the optimum, in the sense that the male curve is monotonically increasing until this point, and the female curve is monotonically decreasing. This λ also corresponds to the reflection through the hyperplane d_{plane} in the direction d_{mean}

These results hold also for other training data and test sets, as presented in the appendix on Figure 11 and Figure 12. In general, it seems that the conclusion to draw is that the gender direction we found does not satisfy all the properties we wanted.

4.3.3 Text generation

For the last experiment, we present text generation from sentences containing gender, and with the intervention on all layers. The table below gives an example of sentence and gpt2-xl's generation. It shows that generation is very fragile and that it is hard to steer the model in general: either nothing happens and the sentence's meaning is not changed, or we are able to steer the model into not using gender or using the opposite gender, but at the cost of making the sentence less meaningful.

This follows logically from earlier experiments: neither are we able to steer the model on questions nor on generation, this means that the direction that we learned doesn't have the expected properties.

<table>
<tr><th colspan="3">Generation</th></tr>
<tr><td></td><td>Lambda</td><td>Sentence</td></tr>
<tr><td>Base</td><td>0</td><td>He is so pretty, I want to ask| him out.</td></tr>
<tr><td>Erase</td><td>1</td><td>He is so pretty, I want to ask| him to marry me.</td></tr>
<tr><td>Steer</td><td>2</td><td>He is so pretty, I want to ask| her to be so pretty.</td></tr>
</table>

I choose to not give more sentences, as the result of this experiment will likely not be very informative.

5 Conclusion

In this report, we have presented the language model gpt2-xl and a method inspired by SOTA research to try and find conceptual direction inside the model's activations. It was motivated by safety issues and my interest in interpreting neural networks to understand their behaviors. I proposed an implementation of the *Leace* technique, and some improvements like *quantile leace*, inspired by [1, 23, 13] to remove and modify gender at inference time.

Using data found online, and generated thanks to the SOTA language model ChatGPT and Claude, I was able to design and run several experiments to test the viability of such a method. The results show that finding the gender direction is harder than we expected. Still, they give us important information:

- Gendered words and names mostly use one direction to differentiate their gender, but this direction is not optimal for applying erasure and steering methods.
- Both Steering and erasure have an effect on the confidence of the model in the actual answer, but this is not enough to change the accuracy and text generated at inference time.

This method is in itself not sufficient but is the first step toward understanding and controlling gender in gpt2-xl.

As a follow-up to this work, one could attempt to use other techniques from circuit discovery to find the important parts of the network computing gender and intervene in them. This would hopefully yield better results in finding or not the gender direction.

Once the present technique works, I would like to test the best setting on a gender benchmark to see a standardized score and compare it to other methods. Another test that I have not been able to perform was to compare the extent to which the gendered word direction could be used to classify/steer/erase the gendered names. This would argue for or against the unicity of a gendered direction as we usually picture it. Hopefully, I will be able to successfully perform these tests in the coming months, and write a workshop/conference paper with my mentor.

References

[1] Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. Leace: Perfect linear concept erasure in closed form, 2023.

[2] Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems* (2016), D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc.

[3] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[4] Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision, 2022.

[5] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[6] Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability, 2023.

[7] Dinan, E., Fan, A., Wu, L., Weston, J., Kiela, D., and Williams, A. Multi-dimensional gender bias classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 314–331.

[8] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. *Transformer Circuits Thread* (2022).

[9] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread* (2021).

[10] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022.

[11] Janus. Simulators. *LessWrong* (Sept 2022).

[12] Langosco, L., Koch, J., Sharkey, L., Pfau, J., Orseau, L., and Krueger, D. Goal misgeneralization in deep reinforcement learning.

[13] Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model, 2023.

[14] Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems 36* (2022).

[15] Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer, 2023.

[16] Nanda, N. Actually, othello-gpt has a linear emergent world model. *neelnanda.io* (Mar 2023).

[17] Nardo, C. The waluigi effect (mega-post). *LessWrong* (Mars 2023).

[18] Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill 5* (03 2020).

[19] Pearl, J. *Causality.* Cambridge University Press, 2009.

[20] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog 1*, 8 (2019), 9.

[21] Rimsky, N. Reducing sycophancy and improving honesty via activation steering. *LessWrong* (July 2023).

[22] Scherrer, N., Shi, C., Feder, A., and Blei, D. M. Evaluating the moral beliefs encoded in llms, 2023.

[23] Turner, A., Monte, M., Udell, D., Thiergart, L., and Mini, U. Steering gpt-2-xl by adding an activation vector. *LessWrong* (May 2023).

[24] Turner, A., Smith, L., Shah, R., Critch, A., and Tadepalli, P. Optimal policies tend to seek power. In *Advances in Neural Information Processing Systems* (2021), M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., pp. 23063–23074.

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[26] Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations* (2023).

A Data Tables

In this appendix, I present the main datasets I used to learn and test the probing, erasing, and steering methods.

Training Set			
	Male word	Female word	Total size
Pronouns	’ He’, ’ His’, ’He’, ’His’, ’ he’, ’ his’, ’he’, ’his’	’ She’, ’ Her’, ’She’, ’Her’, ’ she’, ’ her’, ’she’, ’her’	16
Anatomy	’andropause’, ’glans penis’, ’testosterone’, ’penis’, ’sperm’, ’prostate’, ’urethra’	’menopause’, ’clitoris’, ’estradiol’, ’vagina’, ’ovum’, ’skene gland’, ’uterus’	14
Nouns	’countryman’, ’wizards’, ’manservant’, ’fathers’, ’divo’, ’actor’, ...	’countrywoman’, ’witches’, ’maidservant’, ’mothers’, ’diva’, ’actress’, ...	380
Names	’Emanuel’, ’Leon’, ’Glen’, ’Matt’, ’Antonio’, ’Joshua’, ’Bill’, ’Nicholas’, ...	’Annette’, ’Eugenia’, ’Eula’, ’Gertie’, ’Helen’, ’Margie’, ’Hazel’, ’Gussie’, ...	746

Training prompts		
	Prompts	Size
Pronoun prompts	"The detective examined the crime scene.", "The teacher stood at the front of the classroom.", "The chef prepared a delicious meal.", ...	25
Anatomy prompts	"The doctor told me I have a problem with my ", "The medical term is ", "The ", " "	4
Nouns prompts	"The ", " ", "It doesn’t matter who you are, you can be the ", "My friend often dresses like a ", ...	7
Names prompts	"", "My name is ", "His name is ", "Her name is ".	6

The table Training Set contains examples of the gendered words and names that I used to learn the leace and logistic hyperplanes. They are combined with the Training prompts in the following way: each different pronoun is combined with each different pronoun prompt, and the same applies to anatomy, nouns, and names. Thus, the word dataset contains a total of 3116 unique activation vectors at each layer, and for the names, there is 2280 activation vectors.

Testing Set		
	Prompts	Size
Pre-prompt	"The task is to predict the gender of the character in the following sentences. Answer only using the word 'female' or 'male'. He is a boy. Answer: male She is a girl. Answer female"	2
Female word prompts	"The young lady is talented and is a musician. Answer:", "The actress is skilled and performed flawlessly. Answer:", "The daughter is intelligent and loves to read. Answer:", ...	18
Male word prompts	"The young lord is talented and is a musician. Answer:", "The actor is skilled and performed flawlessly. Answer:", "The son is intelligent and loves to read. Answer:", ...	15
Female names prompts	"Hi, my name is Mary. Answer:", "Hi, my name is Sarah. Answer:", "Hi, my name is Lily. Answer:", ...	37

The table Testing Set contains examples of the prompts I used to test the erasing and steering intervention. A prompt is constructed by concatenation of one of the two pre-prompts with one of the prompts for either names or words. There are two pre-prompts by varying the order of the few-shot learning: one has "He is a boy. Answer: male", and the other starts with "She is a girl. Answer female".

Each prompt was tested such that gpt2-xl always answers the good prediction on the prompts, and such that the influence of the ordering of the few-shot examples is negligible.

B Additional graphs

In the appendix, I present the same graphs as shown in the report, but with other training and test data. In particular, I use the gendered words evaluated on male words and the gendered names evaluated on female names.

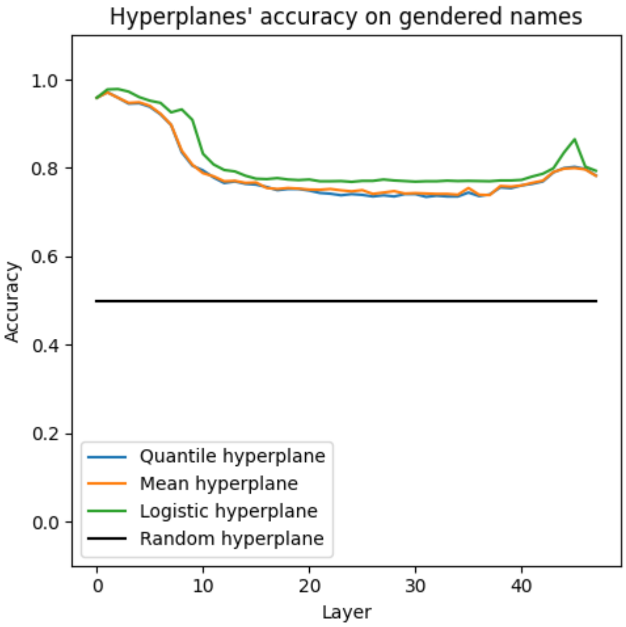

Figure 8: Accuracy of the Logistic regression, the quantile, and the mean hyperplanes measured at each layer in the residual stream. We learned the hyperplanes on all the name training data and tested them on the same data. We observe that gendered names' activations are more easily separable than the gendered words of Figure 4.

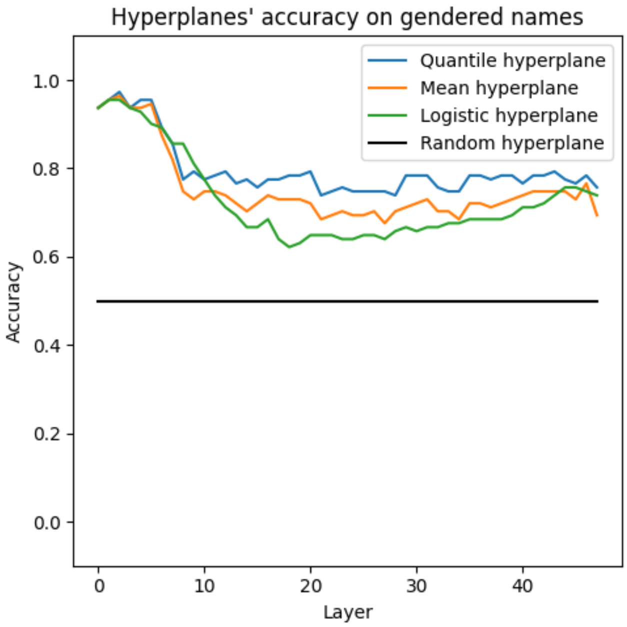

Figure 9: Accuracy of the logistic regression, the quantile, and the mean hyperplanes measured at each layer in the residual stream. We learned the hyperplanes on 90% of the name training data and tested on 10%. Once again the accuracy is lower than in Figure 8, and compared to Figure 5, the quantile hyperplane is here the best. In this case, the logistic regression seems to not generalize well to unobserved names

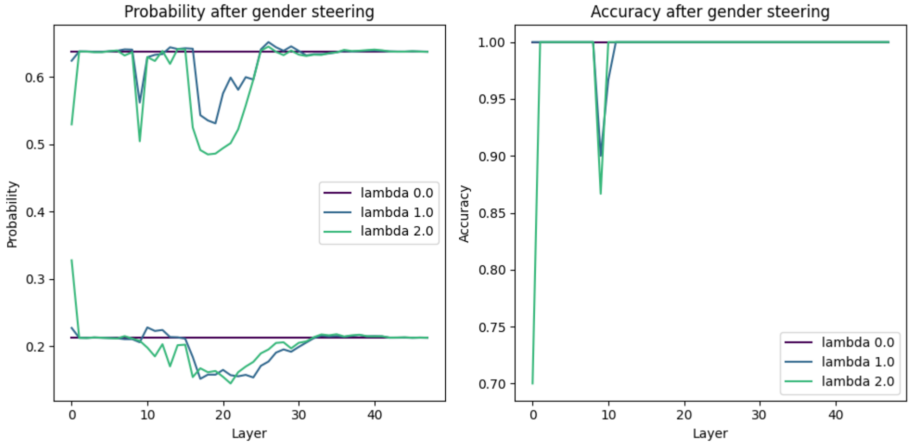

Figure 10: Probability and accuracy of intervention at each layer on male words. Probability and accuracy of intervention at each layer on female words. The lambdas represent different steering coefficients, with $\lambda = 0$, being the non-modified computation, and $\lambda = 1$ being the leace erasure. Contrary to Figure 6, the male gender is harder to steer, and adding the direction only destroys the capability to answer the questions.

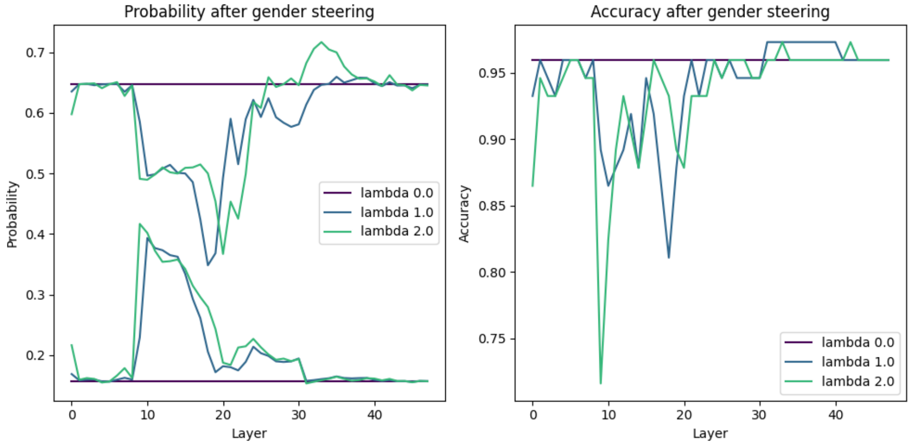

Figure 11: Probability and accuracy of intervention at each layer on female names. Probability and accuracy of intervention at each layer on female words. The lambdas represent different steering coefficients, with $\lambda = 0$, being the non-modified computation, and $\lambda = 1$ being the leace erasure. Observations are similar to Figure 6, but here the probability of the male token increases substantially.

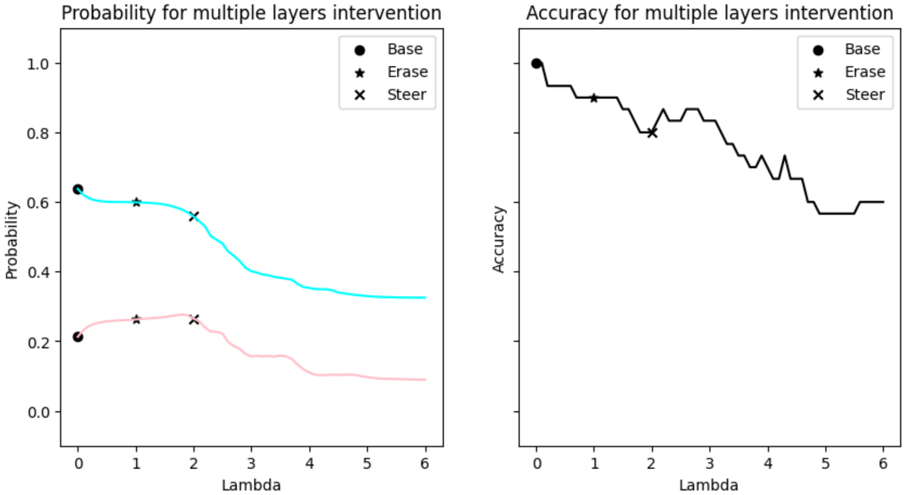

Figure 12: Probability and accuracy when intervening on all layers simultaneously, with leace estimators learned on the gendered words and tested on the male words sentences. The axis Lambda is the amount of steering done at each layer. $\lambda = 0$ is the non-modified computation, and $\lambda = 1$ correspond to the leace erasure. Steering at high λ don't make the probability collapse to 0, which was unexpected.

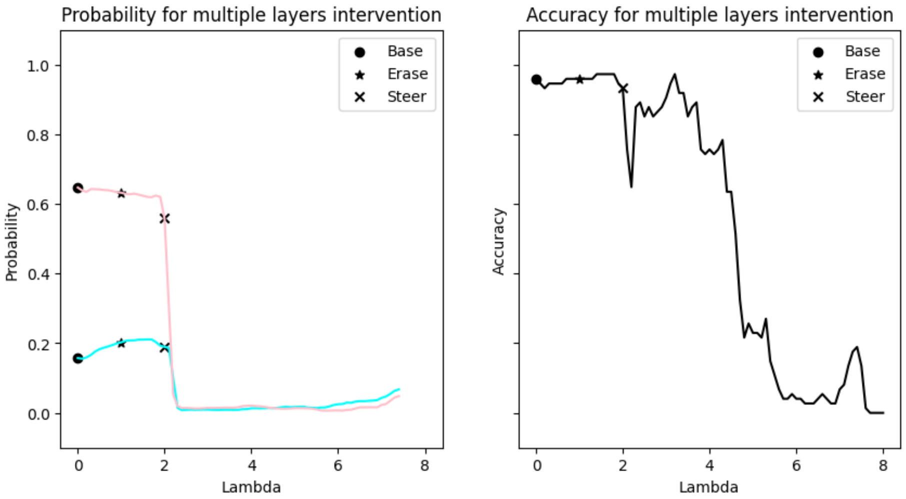

Figure 13: Probability and accuracy when intervening on all layers simultaneously, with leace learned on the gendered names and tested on the female words sentences. The axis Lambda is the amount of steering done at each layer. $\lambda = 0$ is the non-modified computation, and $\lambda = 1$ correspond to the leace erasure. We can observe a very sharp cutoff at $\lambda = 2$, meaning that going further is out-of-distribution for the model.