Sparse Interventions in Language Models with Differentiable Masking

Nicola De Cao [*1,2], **Leon Schmid** [*3], **Dieuwke Hupkes** [4], **Ivan Titov** [1,2,5]

[1]University of Amsterdam, [2]University of Edinburgh
[3]University of Osnabrück, [4]Facebook AI Research, [5]Innopolis University
`nicola.decao@gmail.com, lschmid@uos.de`
`dieuwkehupkes@fb.com, ititov@inf.ed.ac.uk`

Abstract

There has been a lot of interest in understanding what information is captured by hidden representations of language models (LMs). Typically, interpretation methods i) do not guarantee that the model actually uses the information found to be encoded, and ii) do not discover small subsets of neurons responsible for a considered phenomenon. Inspired by causal mediation analysis, we propose a method that discovers a small subset of neurons within a neural LM responsible for a particular linguistic phenomenon, i.e., subsets causing a change in the corresponding token emission probabilities. We use a differentiable relaxation to approximately search through the combinatorial space. An L_0 regularization term ensures that the search converges to discrete and sparse solutions. We apply our method to analyze subject-verb number agreement and gender bias detection in LSTMs. We observe that it is fast and finds better solutions than alternatives such as REINFORCE and Integrated Gradients. Our experiments confirm that each of these phenomena is mediated through a small subset of neurons that do not play any other discernible role.

1 Introduction

The success of language models (LMs) in many natural language processing tasks is accompanied by an increasing interest in interpreting and analyzing such models. One goal in this direction is to identify how a model employs its hidden representations to arrive at a prediction (Belinkov and Glass, 2019; Jacovi and Goldberg, 2020). A popular line of research studies LMs with "diagnostic classifiers" or "probes" that are trained to predict linguistics properties from hidden units, with the purpose of analyzing what information is encoded by the network and where (Alain and Bengio, 2017; Adi et al., 2017; Hupkes et al., 2018; Voita and Titov, 2020).

_____
*Equal contributions.

However, this method is sometimes criticized for generating unfaithful interpretations (Barrett et al., 2019) since the trained classifiers only measure the correlation between a model's representations and an external property and not whether such property is actually causing the model's predictions. Indeed, several studies pointed out limitations of probes (Belinkov and Glass, 2019; Vanmassenhove et al., 2017; Tamkin et al., 2020), including mismatches between the performance of the probe and the original model and the discrepancy between correlation and causation of hidden units and model outputs.

In response to these limitations, several recent studies have proposed to interpret neural models with *interventions* which aim to measure causal effects by intervening in representations of the model and observing a change in the model output (Giulianelli et al., 2018; Elazar et al., 2021; Feder et al., 2021). These techniques investigate directly if an LM represents a certain linguistic phenomenon but are limited when it comes to understanding where and how this information is represented. Therefore, an important question that they cannot answer is to what extent *modularity* – often believed to be a prerequisite for systematic generalization (Goyal and Bengio, 2020; Dittadi et al., 2021) – is a property that emerges naturally in such models. An adaptation of *causal mediation analysis* (Pearl, 2001) by Lakretz et al. (2019); Vig et al. (2020); Lakretz et al. (2021) makes an important step towards enabling such investigations. They consider neurons one by one by setting their activation to zero and measuring their effect on the output. However, these techniques suffer from two major shortcomings: i) they are restricted to detecting single neurons as systematically ablating combinations of multiple neurons is computationally infeasible, and ii) there is no guarantee that setting a unit activation to zero corresponds to switching the corresponding function on or off (Sturmfels et al., 2020).

Here, we use a differentiable relaxation of this

search problem to overcome both these limitations. More specifically, our goal is to identify neurons responsible for shifting the probability from a word to its alternative in examples exemplifying the phenomena, without affecting other LM predictions. For example, when investigating subject-verb number agreement, we want to redistribute the probability mass from the singular form of an upcoming verb to the plural one (or vice versa), while discouraging changes in the distributions for other contexts. In this way, we ensure that the function is mediated through the detected neurons, and these neurons do not play any other discernible role.

Building on the framework of differentiable masking (De Cao et al., 2020; Schlichtkrull et al., 2021), we formalize this search for a *sparse intervention* as a constrained optimization problem. We aim to both detect the responsible neurons and learn the values to assign them when intervening. We use a continuous relaxation of the subset-selection problem, but ensure discreteness and encourage sparsity through L_0 regularization. The L_0 penalty determines how many neurons we want to discover. In our experiments, we use an LSTM-based LM, previously investigated by (Gulordava et al., 2018; Lakretz et al., 2019), and consider subject-verb number agreement and gender bias detection. We start with validating our method by showing that we can replicate findings reported in these previous studies and then dive into a deeper analysis. We show that our proposed method is effective as well as computationally efficient – it converges up to 7 times faster than REINFORCE (Williams, 1992) and surpasses Integrated Gradients (Sundararajan et al., 2017) in terms of accuracy/sparsity.

2 Related Work

The L_0 regularization was proposed by Louizos et al. (2018) in the context of pruning neural network weights and biases. It has been used in a variety of works in NLP as a tool for generating rationales and attribution (Bastings et al., 2019; De Cao et al., 2020; Schlichtkrull et al., 2021). Masking weights and groups of weights was also used by Csordás et al. (2021) to investigate the functional modularity of neural networks.

Studies suggested that some of the linguistic phenomena are encoded, at least to a large degree, in a disentangled and sparse fashion. For example, Radford et al. (2017) detected a neuron encoding sentiment polarity and Dai et al. (2021) showed

that individual facts learned by an LM can be manipulated by modifying a small number of neurons. In a similar spirit, Voita et al. (2019) observed that many Transformer attention heads in a neural machine translation model are specialized; interestingly, they also used L_0 regularization but only to prune less important heads; the roles played by the heads were identified by studying their attention patterns. Our technique can facilitate such studies by effectively identifying sets of neural network's subcomponents playing a given function.

Bau et al. (2019) use different kinds of correlations between neurons from different models to measure their importance. The authors find that many individual neurons capture common linguistic phenomena, also showing how to control translations in predictable ways by modifying their activations. Similarly to Lakretz et al. (2021), the work of Finlayson et al. (2021) instead focuses on models' preferences for grammatical inflections, as well as whether neurons process subject-verb agreement. The authors include causal mediation analysis in their methodology.

Conversely, Antverg and Belinkov (2022) criticize recently proposed methodologies for analyzing individual neurons in LMs. In particular, they discuss methods that rely on an external probe to rank neurons according to their relevance to some linguistic attribute. They indicate two main pitfalls: 1) these methodologies confound probe quality and ranking quality, and 2) they focus on encoded information rather than information that the model uses. Their analysis does not apply to ours since we do not use probes explicit.

Finally, we refer the reader to Sajjad et al. (2021) for a recent survey of neuron-level interpretation of NLP models, which includes methods to discover neurons, evaluation methods, significant findings and future research directions.

3 Method

We investigate if we can find groups of neurons for which a modification of their value – which we call an *intervention* – systematically leads to a change of probability for the single token emission related to a specific phenomenon. Because there is no direct supervision for interventions, we need to learn them with a proxy objective. Let's assume we have an autoregressive model (e.g., an LSTM; Hochreiter and Schmidhuber 1997) that assigns a probability to sequences. For a set of input tokens

$x = \langle x_1, \ldots, x_n \rangle$, we obtain the model's probability of the token of interest $p(x_n|x_{<n})$ along with the hidden states $h = \langle h_1, \ldots, h_n \rangle$ where $h_i \in \mathbb{R}^k$ (one for each time step). We then intervene in the model's computation by modifying a group of neurons from one or multiple hidden states. The intervention at a certain time step $i < n$ consists of a binary mask $m \in \{0, 1\}^k$ indicating which hidden units need intervention and which can be left unchanged. The intervention is then made substituting the ith hidden state with the altered state

$$\hat{h}_i = (1 - m) \odot h_i + m \odot b \,, \qquad (1)$$

where $\odot$ indicates the element-wise product and $b \in \mathbb{R}^k$ is a learned baseline vector that will lead the desired intervention. We denote $p(x_n|x_{<n}, \hat{h}_i)$ as the model's probability of the target token when its forward pass has been altered using $\hat{h}_i$.

In addition, as the main objective of this work, we are looking for sparse interventions, which we define as finding a defined small percentage (e.g., 1-5%) of neurons where to apply an intervention to while keeping all the rest untouched.

3.1 Learning to Intervene

Because there is no direct supervision to estimate the mask m and the baseline b, we minimize

$$\mathcal{L}_{\text{ratio}}(\hat{h}_i, x) = \frac{p(x_n = d|x_{<n}, \hat{h}_i)}{p(x_n = t|x_{<n}, \hat{h}_i)} \,, \qquad (2)$$

where we want to identify neurons responsible for a change in probability between a predicted word d and a target word t (e.g., a singular and plural verb form—where, independently from which form is correct, d is the form that the model assigns the highest probability to, and t to the other). In other words, we optimize to assign more probability mass to the token t rather than d. In addition, we desire interventions to be as sparse as possible, because we want to identify the least number of neurons responsible for the decision. Such *sparsity* corresponds to constraining most of the entries of m to be 0, which corresponds to not interfering. We cast this in the language of constrained optimization.

A practical way to express the sparsity constraint is through the L_0 'norm'. Our constraint is defined as the total number of neurons we intervene on:

$$\mathcal{C}_0(m) = \sum_{i=1}^{k} \mathbf{1}_{[\mathbb{R}_{\neq 0}]}(m_i) \,. \qquad (3)$$

The whole optimization problem is then:

$$\min_{m,b} \sum_{x \in \mathcal{D}} \mathcal{L}_{\text{ratio}}(\hat{h}_i, x) \ \text{ s.t. } \mathcal{C}_0(m) \leq \alpha \,, \qquad (4)$$

where $\mathcal{D}$ is a dataset and the margin $\alpha \in (0, 1]$ is a hyperparameter that controls the desired sparsity (i.e., the lower α, the sparser the solution will be). Since non-linear constrained optimization is generally intractable, we employ Lagrangian relaxation (Boyd et al., 2004) optimizing

$$\max_{\lambda} \min_{m_i, b} \sum_{x \in \mathcal{D}} \mathcal{L}_{\text{ratio}}(\hat{h}_i) + \lambda(\mathcal{C}_0(m_i) - \alpha) \,, \quad (5)$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is the Lagrangian multiplier. Since we use binary masks, our loss is discontinuous and non-differentiable. A default option would be to use REINFORCE (Williams, 1992), but it is known to have a noisy gradient and thus slow convergence. To overcome both problems, we resort to a sparse relaxation to binary variables, namely using a Hard Concrete distribution (Louizos et al., 2018) (see Section 3.5 for more details).

3.2 Stochastic relaxation of the Mask

Our optimization problem poses two difficulties: i) $\mathcal{C}_0$ is discontinuous and has zero derivative almost everywhere, and ii) the altered state $\hat{h}_i$ is discontinuous w.r.t. the binary mask m. A simple way to overcome both issues is to treat the binary mask as stochastic and optimize the objective in expectation. In that case, one natural option is to resort to score function estimation (REINFORCE; Williams, 1992) while another is to use a sparse relaxation to binary variables (Louizos et al., 2018; Bastings et al., 2019; De Cao et al., 2020; Schlichtkrull et al., 2021). In Section 4 we discuss the two aforementioned options showing that the latter is much more effective (results in Table 6). Thus we opt to use the Hard Concrete distribution, a mixed discrete-continuous distribution on the closed interval $[0, 1]$. This distribution assigns a non-zero probability to exactly zero and one while it also admits continuous outcomes in the unit interval via the *reparameterization trick* (Kingma and Welling, 2014). We refer to Louizos et al. (2018) for details, but also provide a brief summary in Section 3.5. With a stochastic mask, the objective is computed in expectation, which addresses both sources of non-differentiability:

$$\mathcal{C}_0(m) = \sum_{i=1}^{k} \mathbb{E}_{p(m_i)}[m_i \neq 0] \,. \qquad (6)$$

Note that during training the mask is sampled and its values lies in the closed unit interval. After training, we set the mask entries to exact ones when their expected values are > 0.5 or to zero otherwise. To prevent issues due to the discrepancy between the values of the mask during training and during inference, we add another constraint

$$\mathcal{C}_{(0,1)} = \sum_{i=1}^{k} \mathbb{E}[m_i \in (0,1)] \,, \quad (7)$$

to be $\leq \beta$. $\mathcal{C}_{(0,1)}$ during training constrains the relaxed mask values not to lie in the open interval $(0,1)$ but rather to concentrate in $\{0,1\}$. $\beta \in (0,1]$ is a hyperparameter (the lower the less discrepancy is expected).

3.3 Single-step and Every-step intervention

We described how we apply an intervention at a certain time step $i < n$ as an intervention that directly modifies h_i. We refer to this type as a *single-step* intervention. The choice of the time step to intervene should be carefully set to investigate a particular phenomenon in the LM, and is task dependent; e.g., to explore subject-verb agreement, a reasonable choice is to do the intervention at the hidden state of the subject. As an extension, we also define an *every-step* intervention when instead of altering only h_i we modify all $h_1, \dots, h_{n-1}$ with the same m and b (similar to Lakretz et al. 2019). The two types of intervention investigate different properties of an LM; we experiment with both variants.

3.4 Retaining other predictions

We train interventions to modify the model's prediction at a specific token position. However, there is little guarantee that all the other token positions will have the same output distribution as without the interventions. This is important as, when investigating modularity, we would like to ensure not only that a group of neurons plays a distinct interpretable role but also that they do not fulfil any other discernable role. For this reason, we employ a regularization term in addition to the constrained objective. This corresponds to minimizing a Kullback–Leibler divergence between the output distributions of the original model and the one from the model with interventions. The regularization term is a KL divergence between the output distributions of the original model p_O and the one from the model with interventions p_I averaged at every token position: $\mathcal{L}_{KL} =$

$$\frac{1}{T} \sum_{t=1}^{T} D_{KL} \left(p_O(x_t|x_{<t}) \parallel p_I(x_t|x_{<t}) \right) \quad (8)$$

We sum $\mathcal{L}_{KL}$ to Equation 5 multiplied by a factor. This factor is a hyperparameter that controls the amount of regularization to apply, and we empirically found that 1.0 is a good value. In practice, as we will discuss in Section 5, the regularization term does not play an important role.

3.5 The Hard Concrete distribution

The Hard Concrete distribution, assigns density to continuous outcomes in the open interval $(0,1)$ and non-zero mass to exactly 0 and exactly 1. A particularly appealing property of this distribution is that sampling can be done via a differentiable reparameterization (Rezende et al., 2014; Kingma and Welling, 2014). In this way, the $\mathcal{C}_0$ constrain in Equation 3 becomes an expectation (Equation 6) whose gradient can be estimated via Monte Carlo sampling without the need for REINFORCE and without introducing biases. We did modify the original Hard Concrete, though only so slightly, in a way that it gives support to samples in the half-open interval $[0,1)$, that is, with non-zero mass only at 0. That is because we need only distinguish 0 from non-zero, and the value 1 is not particularly important.[1]

The distribution A stretched and rectified Binary Concrete (also known as Hard Concrete) distribution is obtained applying an affine transformation to the Binary Concrete distribution (Maddison et al., 2017; Jang et al., 2017) and rectifying its samples in the interval $[0,1]$. A Binary Concrete is defined over the open interval $(0,1)$ and it is parameterised by a location parameter $\gamma \in \mathbb{R}$ and temperature parameter $\tau \in \mathbb{R}_{>0}$. The location acts as a logit and it controls the probability mass skewing the distribution towards 0 in case of negative location and towards 1 in case of positive location. The temperature parameter controls the concentration of the distribution. The Binary Concrete is then stretched with an affine transformation extending its support to (l, r) with $l \leq 0$ and $r \geq 1$. Finally, we obtain a Hard Concrete distribution rectifying samples in the interval $[0,1]$. This corresponds to

---

[1]Only a true 0 is guaranteed to completely mask an input out, while any non-zero value, however small, may leak some amount of information.

collapsing the probability mass over the interval $(l, 0]$ to 0, and the mass over the interval $[1, r)$ to 1. This induces a distribution over the close interval $[0, 1]$ with non-zero mass at 0 and 1. Samples are obtained using

$$\begin{aligned} s &= \sigma\left(\left(\log u - \log(1 - u) + \gamma\right)/\tau\right) , \\ z &= \min\left(1, \max\left(0, s \cdot (l - r) + r\right)\right) , \end{aligned} \quad (9)$$

where σ is the Sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$ and $u \sim \mathcal{U}(0, 1)$. We point to the Appendix B of Louizos et al. (2018) for more information about the density of the resulting distribution and its cumulative density function.

4 Experimental Setting

We study the pre-trained LSTM language model made available by Gulordava et al. (2018)[2], which has been studied extensively in previous works and therefore provides a good testing ground for our method. The studied model is a standard two-layered LSTM with a hidden dimension of 650. The embedding layer also has dimensionality 650, and it is not tied with the output layer. The vocabulary size is $50,000$ and the model was trained on English Wikipedia data (with around 80M tokens training tokens and 10M for validation). We used this model to compare to previous findings of Lakretz et al. (2019). We also pre-train this LM several times with different weights initializations to make sure our results generalize.

We study the original model, as well as newly trained models with the same architecture, on two tasks described below: subject-verb number agreement and gender bias. The evaluation for tasks naturally follows the defined objective $\mathcal{L}_{\text{ratio}}(\hat{h}_i, x)$ (see § 3.1). Without intervention, the ratio is always > 1. Thus, we define a successful intervention when we find a mask and baseline values such that the ratio becomes < 1. Then, we define the *accuracy* of interventions as the average number of times that the ratio is < 1 across all datapoints in a given dataset/task. The accuracy thus reflects how often we can *flip* the model's decision.

Subject-verb number agreement Here, we seek the neurons responsible for predicting the number of verb forms: for a given sentence, we wish the intervention to change the number of the verb from singular to plural or vice versa. For this

task, we employ data made available by Lakretz et al. (2019)[3]. The data are synthetic and generated with a modified version from Linzen et al. (2016) and Gulordava et al. (2018). Each synthetic number-agreement instance has a fixed syntax and varied lexical material. Sentences were randomly sampled by choosing words from pools of 20 subject/object nouns, 15 verbs, 10 adverbs, 5 prepositions, 10 proper nouns and 10 location nouns. We used a total of 11,000 training sentences and 1,000 evaluation sentences. We apply the single-step intervention to the subject of the (only) verb. We apply two intervention here (i.e., two sets of mask and baseline values): one where we train the model to turn the verb into the singular form and one into the plural one.

Gender bias detection In this task, we seek the neurons responsible for setting pronoun genders: for a given sentence, we wish the intervention to change the pronoun that refers to a person with a profession and an unspecified gender. For this task, we employ data made available by Vig et al. (2020)[4]. The data are synthetic and generated with a list of templates from Lu et al. (2020) and several other templates, instantiated with professions from Bolukbasi et al. (2016) (17 templates and 169 professions, resulting in 2,873 examples in total). We refer to Vig et al. (2020) for the full lists of templates and professions. The templates have the form "The [occupation] [verb] because [he/she]". Professions are definitionally gender-neutral. We used a total of 2,673 training sentences and 200 evaluation sentences. Also for his task, we apply the single-step intervention to the subject of the sentence, using different interventions for flipping the pronoun to "*he*" and to "*she*".

5 Results

For the single-step intervention (with regularization), our method achieves 91.5 and 93.9 accuracies for the number agreement and gender bias tasks, respectively. On average, our method finds 5.7 and 5.3 units for the two tasks, respectively. Considering that the LM has 1,300 hidden units, this intervention is relatively sparse as desired (we use $< 0.41\%$ of the total units). In Figure 1 and 2, we show examples of hidden state activations with

---

[2] https://github.com/facebookresearch/colorlessgreenRNNs

[3] https://github.com/FAIRNS/Number_and_syntax_units_in_LSTM_LMs
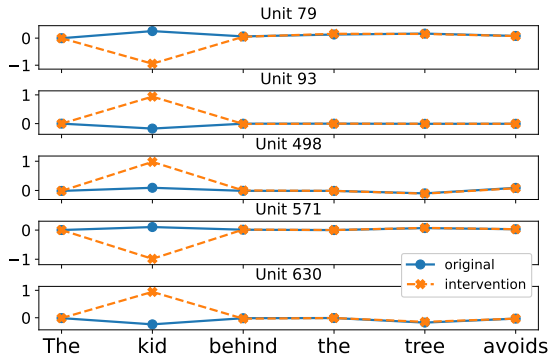[4] https://github.com/sebastianGehrmann/CausalMediationAnalysis

Figure 1: Activations of four units we intervene on (single step intervention at "kid") for changing number agreement (at "avoids").
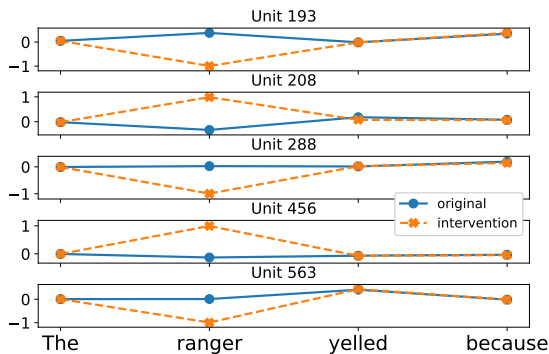

Figure 2: Activations of four units we intervene on (single step intervention at "ranger") for changing the pronoun (after "because").

Unit	Singular	Plural	Prevalence
79	-0.96 $_{\pm 0.02}$	0.99 $_{\pm 0.01}$	100%
93	0.95 $_{\pm 0.03}$	-0.84 $_{\pm 0.09}$	100%
243	0.91 $_{\pm 0.06}$	0.18 $_{\pm 0.15}$	20%
357	-0.99 $_{\pm 0.01}$	0.87 $_{\pm 0.03}$	40%
498	0.98 $_{\pm 0.01}$	-0.96 $_{\pm 0.03}$	100%
571	-0.99 $_{\pm 0.01}$	0.93 $_{\pm 0.06}$	80%
630	0.95 $_{\pm 0.03}$	0.11 $_{\pm 0.26}$	100%
776	-0.81 $_{\pm 0.05}$	0.96 $_{\pm 0.01}$	20%
988	1.00 $_{\pm 0.00}$	-0.99 $_{\pm 0.00}$	10%

Table 1: Subject-verb number agreement task with single-step interventions. Values are averages across 10 runs.

Unit	He	She	Prevalence
193	-0.99 $_{\pm 0.00}$	0.91 $_{\pm 0.01}$	100%
208	0.99 $_{\pm 0.00}$	-0.96 $_{\pm 0.01}$	100%
288	-0.99 $_{\pm 0.00}$	-0.47 $_{\pm 0.14}$	100%
455	-0.99 $_{\pm 0.00}$	0.10 $_{\pm 0.01}$	20%
456	0.99 $_{\pm 0.00}$	-0.98 $_{\pm 0.00}$	100%
513	0.98 $_{\pm 0.00}$	-0.74 $_{\pm 0.00}$	10%
563	-0.99 $_{\pm 0.00}$	0.96 $_{\pm 0.01}$	100%

Table 2: Gender bias task with single-step interventions. Values are averages across 10 runs.

and without interventions for both tasks (see Appendix A for additional examples). From these figures, we can see that only one time-step is heavily affected (the one of the intervention) while the others are minimally corrupted after that time step. We hypothesize that the model stores the information of number or gender in other units (or in cell states), but the discovered units are the ones responsible for the *initialization* of such memory. In Table 1 and 2, we report the full list of discovered units and the learned baseline vectors for both tasks on the single-step intervention.

For the every-step intervention, our method achieved an almost perfect accuracy of 95.8 and 99.9 for the number agreement and gender bias tasks, respectively, while using 3 units or less on average for both tasks. This type of intervention is much more effective and more intrusive—the number of changes is larger as it happens at every step). In Table 3 we report the full list of discovered units and the learned baseline vectors, comparing to the one discovered by Lakretz et al. (2019) (every-step

intervention). Noticeably, we *re-discover* unit 776 which validates our method and confirm their findings. Interestingly, we also discover an extra unit on average, highlighting that one of the limitations of Lakretz et al. (2019) was indeed an efficient way to search units. For a summary of all results see Table 4, and for the discovered units and baseline on the gender task see Table 5.

Efficiency To demonstrate the efficiency and efficacy of our estimation employing the Hard Concrete distribution, we compare to the standard Score Function Estimation (aka REINFORCE; Williams 1992) with a moving average baseline for variance reduction (Botev and Ridder, 2017) and trying different values of α to achieve a good trade-off between accuracy and number of units used. We also compare to Integrated Gradients (Sundararajan et al., 2017) where we intervene on the top-k influential neurons by setting them to zero. In Table 6, we summarize the results for the single-step intervention. REINFORCE takes at least 7 times more time to converge, and it always converges at using more units than our method with

Unit	Singular	Plural	Prevalence	Found by Lakretz et al. (2019)
79	-0.76 $_{\pm 0.023}$	0.99 $_{\pm 0.003}$	100%	✗
776	-0.99 $_{\pm 0.002}$	0.99 $_{\pm 0.002}$	100%	✓

Table 3: Subject-verb number agreement task with every-step interventions. Values are averages across 10 runs. "Found" indicates how many times our model decides to apply the intervention on a specific unit across runs.

	Accuracy	Units	KL
	Number agreement		
Single	90.9 $_{\pm 1.2}$	5.7 $_{\pm 0.5}$	0.034 $_{\pm 0.034}$
Single^R	91.5 $_{\pm 0.7}$	5.7 $_{\pm 0.9}$	0.035 $_{\pm 0.006}$
Every	96.8 $_{\pm 0.6}$	2.0 $_{\pm 0.0}$	0.131 $_{\pm 0.003}$
Every^R	95.8 $_{\pm 0.4}$	2.0 $_{\pm 0.0}$	0.084 $_{\pm 0.002}$
	Gender bias		
Single	93.1 $_{\pm 4.6}$	5.4 $_{\pm 1.1}$	0.009 $_{\pm 0.001}$
Single^R	93.9 $_{\pm 3.7}$	5.3 $_{\pm 0.5}$	0.009 $_{\pm 0.001}$
Every	98.3 $_{\pm 2.8}$	3.4 $_{\pm 0.5}$	0.176 $_{\pm 0.022}$
Every^R	99.9 $_{\pm 0.3}$	3.0 $_{\pm 0.0}$	0.117 $_{\pm 0.004}$

Table 4: Summary of results for both the number agreement and gender bias settings (average across 3 run for each setting). ^{R} indicates KL regularization. Single/ Every indicates single-step and every-step interventions respectively.

Unit	He	She	Prevalence
288	-0.98 $_{\pm 0.00}$	0.53 $_{\pm 0.05}$	100%
456	0.98 $_{\pm 0.00}$	-0.98 $_{\pm 0.01}$	100%
1184	-0.98 $_{\pm 0.00}$	0.99 $_{\pm 0.00}$	100%

Table 5: Gender bias task with every-step interventions. Values are averages across 10 runs.

lower accuracy. Note that doing an exact search for this problem has a time complexity of $O(2^k)$ where k is the number of neurons—this would amount to $> 10^{12}$ evaluations only for checking combinations up to 4 neurons.

Robustness   To demonstrate that our method is robust, we tested it on 5 language models initialized with different seeds and trained with the original script by Gulordava et al. (2018). We run our method for the single-step intervention 3 times for each language model. The average accuracy at convergence is 88.7 ± 2.6, and we discover 4.7 ± 0.5 units on average. The variability in both accuracy and number of units is very low, indicating that our

	Acc. (↑)	Units (↓)	Speed (↓)
SFE ($\alpha = 0.05$)	**100.0**	20.0	5.2h
SFE ($\alpha = 0.02$)	87.6	6.0	3.6h
IG ($\alpha = 0.005$)	22.5	7.0	–
IG ($\alpha = 0.01$)	28.1	13.0	–
IG ($\alpha = 0.02$)	31.5	26.0	–
Ours ($\alpha = 0.02$)	91.5	**5.7**	**0.5h**

Table 6: Comparison between the solutions found by Score Function Estimation (SFE aka REINFORCE), Integrated Gradients (IG; Sundararajan et al., 2017), and our system (average across 10 runs on a single GPU device). Ours is much faster and finds a sparser solution with better accuracy.

method is robust to different parameterizations of the model we applied it to.

Effect of Regularization   We ablated the KL regularization to see whether it affects learning and the final convergence of our method. On the number agreement task, we found that the average KL divergence with respect to the original model predictions was $0.035/0.084$ with regularization and $0.034/0.131$ without regularization (for single-step and every-step intervention, respectively). We used different regularization coefficients (i.e., different weights), but we did not observe a substantial change in the convergence of our models. Moreover, the accuracy and the number of units found with regularization was almost the same as without regularization (see Table 4 for all results). This lack of effect of the regularization suggests the studied phenomenon is naturally captured by specialized neurons. In the gender bias task, regularization has a similar and negligible impact. The regularized method converges to finding fewer units on average and with worse accuracy (95.8 as opposed to 98.6) in the single-step intervention. In the every-step intervention, the accuracy stays invariant (for both settings is 100) while the model converges to using more units.

6 Conclusions

In this work, we present a new method that employs constraint optimization to efficiently find hidden units that are responsible for particular language phenomena of language models. We use an L_0 regularization to find a sparse solution—., our methodology discovers few units in the order of 2-6 that is $< 0.41\%$ of all units in the studied LM. We show such sparse solutions can be found for multiple phenomena (number and gender agreement) and is an useful tool for analysis of what a LM has learned and how units influence its token emissions. Although this work focuses on LSTM models, the proposed technique is not architecture-dependent and thus easily applicable to transformers, convolution-based models and many others.

Acknowledgements

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *International Conference on Learning Representations*.

Omer Antverg and Yonatan Belinkov. 2022. On the pitfalls of analyzing individual neurons in language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. 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 6330–6335, Hong Kong, China. Association for Computational Linguistics.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4349–4357.

Zdravko Botev and Ad Ridder. 2017. Variance reduction. *Wiley statsRef: Statistics reference online*, pages 1–6.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *ArXiv preprint*, abs/2104.08696.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, Online. Association for Computational Linguistics.

Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wuthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. 2021. On the transfer of disentangled representations in realistic settings. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal Model Explanation Through Counterfactual Language Models. *Computational Linguistics*, 47(2):333–386.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Anirudh Goyal and Yoshua Bengio. 2020. Inductive biases for deep learning of higher-level cognition. *ArXiv preprint*, abs/2011.15091.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, 213:104699. Special Issue in Honour of Jacques Mehler, Cognition's founding editor.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender Bias in Neural Natural Language Processing. *Logic, Language, and Security*, pages 189–202.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Judea Pearl. 2001. Direct and Indirect Effects. *Uncertainty in Artificial Intelligence*, page 411–420.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *ArXiv preprint*, abs/1704.01444.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing,*

China, 21-26 June 2014, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level interpretation of deep NLP models: A survey. *CoRR*, abs/2108.13138.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2021. Interpreting graph neural networks for NLP with differentiable edge masking. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2020. Visualizing the Impact of Feature Attribution Baselines . *Distill*, 5(1):e22.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. Investigating transferability in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1393–1401, Online. Association for Computational Linguistics.

Eva Vanmassenhove, Jinhua Du, and Andy Way. 2017. Investigating 'Aspect' in NMT and SMT: Translating the English simple past and present perfect. *Computational Linguistics in the Netherlands Journal*, 7:109–128.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.
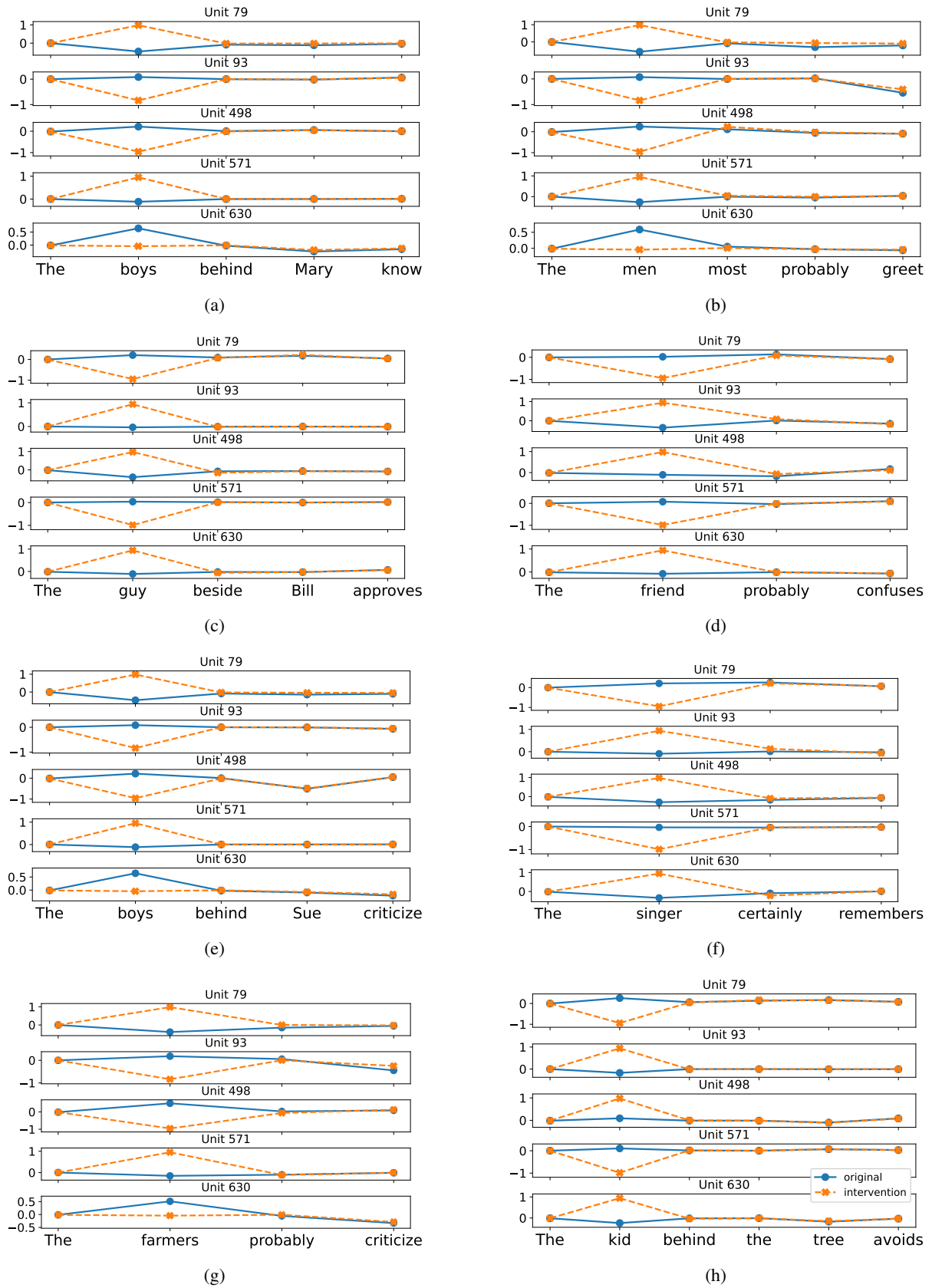
A Additional results

Figure 3: Subject-verb number agreement: activations of four units we intervene on (single step intervention at the second token from the left) for changing number agreement (at the last token).
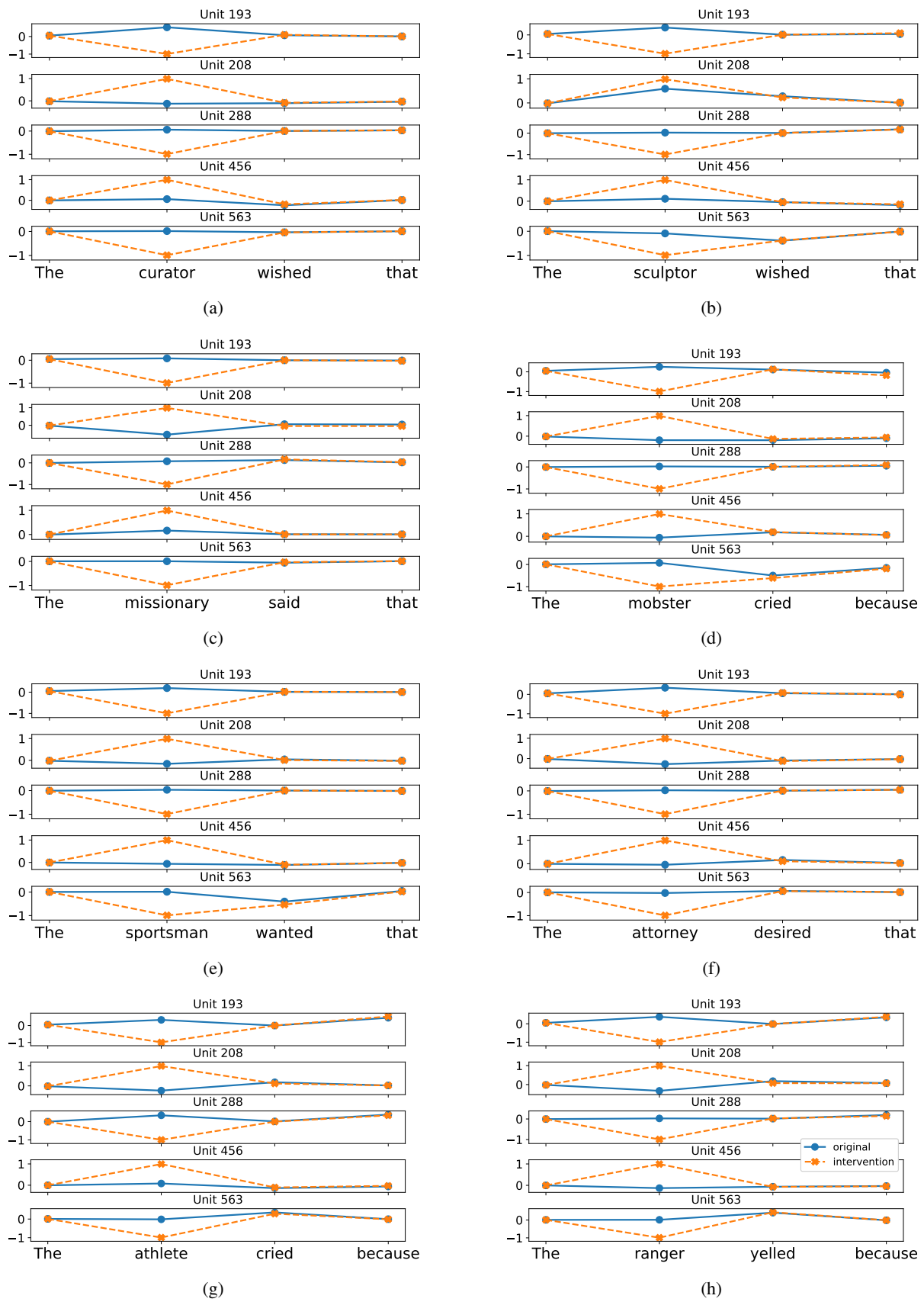
Figure 4: Gender bias: activations of four units we intervene on (single step intervention at the second token from the left) for changing the pronoun (after "because" or "that").