Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction

Shauli Ravfogel[*1,2] **Grusha Prasad**[*3] **Tal Linzen**[4] **Yoav Goldberg**[1,2]
[1]Computer Science Department, Bar Ilan University
[2]Allen Institute for Artificial Intelligence
[3]Cognitive Science Department, Johns Hopkins University
[4]Department of Linguistics and Center for Data Science, New York University
linzen@nyu.edu, grusha.prasad@jhu.edu
{shauli.ravfogel, yoav.goldberg}@gmail.com

Abstract

When language models process syntactically complex sentences, do they use their representations of syntax in a manner that is consistent with the grammar of the language? We propose AlterRep, an intervention-based method to address this question. For any linguistic feature of a given sentence, AlterRep generates counterfactual representations by altering how the feature is encoded, while leaving intact all other aspects of the original representation. By measuring the change in a model's word prediction behavior when these counterfactual representations are substituted for the original ones, we can draw conclusions about the causal effect of the linguistic feature in question on the model's behavior. We apply this method to study how BERT models of different sizes process relative clauses (RCs). We find that BERT variants use RC boundary information during word prediction in a manner that is consistent with the rules of English grammar; this RC boundary information generalizes to a considerable extent across different RC types, suggesting that BERT represents RCs as an abstract linguistic category.

1 Introduction

The success of neural language models, both in NLP tasks and as cognitive models, has fueled targeted evaluation of these models' word prediction accuracy on a range of syntactically complex constructions (Linzen et al., 2016; Gauthier et al., 2020; Warstadt et al., 2020; Mueller et al., 2020; Marvin and Linzen, 2018). What are the internal representations that support such sophisticated syntactic behavior? In this paper, we tackle this question using an intervention-based approach (Woodward, 2005). Our method, AlterRep, is designed to study whether a model uses a particular linguistic feature in a manner which is consistent with the grammar of the language. The method involves two steps: first, it generates *counterfactual*[1] contextual word representations by altering the neural network's representation of the linguistic feature under consideration; and second, it characterizes the change in the model's word prediction behaviour that results from replacing the original word representations with their counterfactual variants. If the resulting change in word prediction aligns with predictions from linguistic theory, we can infer that the model uses the feature under consideration in a manner consistent with the grammar of the language.

We demonstrate the utility of AlterRep using relative clauses (RCs). According to the grammar of English, to correctly determine whether the masked verb in (1) should be singular or plural, a model must recognize that the masked verb is outside the RC *the officers love*, and should therefore agree with the subject of the main clause (*the skater*, which is singular), rather than with the subject of the RC (*the officers*, which is plural).

(1) The skater **the officers love** [MASK] happy.

To investigate whether a neural model uses RC boundary representations as predicted by the grammar of English, we use AlterRep to generate two counterfactual representations of the masked verb: one which encodes (incorrectly) the verb is *inside* the RC, and another which encodes (correctly) that the verb is *outside* the RC. Crucially, the difference between the counterfactual and original representations is *minimal*: the aspects of the representation which do not encode information about RC boundaries remain unchanged. Therefore, if the model uses RC boundary information as dictated by the grammar of English—and if our method successfully identifies the way in which RC boundary information is represented by the model—we expect

[*]Equal contribution.

[1]We use the word *counterfactual* as it is used when referring to *counterfactual examples* (Verma et al., 2020): an altered version of an element that is similar to the original element in all aspects except one.

Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL), pages 194–209
November 10–11, 2021.

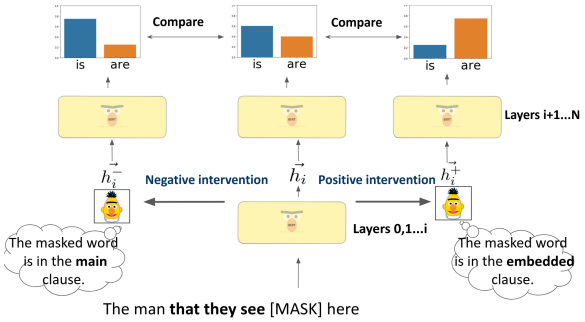


Figure 1: Causal analysis with counterfactual intervention. Given a representation h of a masked word, we derive two new representations h^-, h^+ that differ in the information they contain with respect to a specific linguistic property. The predictions of the model over the counterfactual representations are compared with the original prediction $\hat{Y}$.

the incorrect counterfactual to cause the masked verb to incorrectly agree with the noun *inside* the RC, and the correct counterfactual to cause agreement with the noun *outside* the RC, correctly.

We report experiments applying this logic to BERT variants of different sizes (Devlin et al., 2019; Turc et al., 2019). We found that while all layers of the BERT variants encoded information about RC boundaries, only the information encoded in the middle layers was *used* in a manner consistent with the grammar of English. This contrast highlights the pitfalls of drawing behavioral conclusions from probing results alone, and motivates causal approaches such as AlterRep.

For BERT-base, we also found that counterfactual representations learned solely from one type of RC influenced the model's predictions in sentences containing *other* RC types, suggesting that this model encodes information about RC boundaries in an abstract manner that generalizes across different RC types. Going beyond our case study of RC representations in BERT variants, we hope that future work can apply this method to test linguistically motivated hypotheses about a wide range of structures, tasks and models.

2 Background

2.1 Relative clauses (RCs)

An RC is a subordinate clause that modifies a noun. The head of the RC needs to be interpreted twice—once in the main clause, and once inside the RC—but it is omitted from inside the RC, replaced by an unpronounced "gap". For example, in (2), the RC (in bold) describes the subject of the main clause *the book*. Since *the book* is the object of the embedded clause, we say that the gap is in the object position of the RC (indicated by underscores).

(2) The books **that my cousin likes** ___ were interesting. (Object RC)

RCs can structurally differ from the Object RC in (2) in several ways: the overt complementizer *that* can be excluded, as in (3); the gap can be in the subject instead of object position of the embedded clause, as in (4); and so on. The five types of RCs we consider in this paper are outlined in Table 1.

(3) The books **my cousin likes** ___ were interesting. (Reduced Object RC)

(4) My cousin **that** ___ **likes the books** was interesting. (Subject RC)

These differences do not affect the strategy that a system that follows the grammar of English should use to determine the number of the verb: regardless of the internal structure of the RC, a verb outside the RC should agree with the subject of the main clause, whereas a verb inside the RC should agree with the subject of the RC. Thus, a model that does not properly identify the boundaries of the RC will often predict a singular verb where a plural one is required, or vice versa.

2.2 Iterative Null Space Projection (INLP)

INLP (Ravfogel et al., 2020) is a method for selectively identifying and removing user-defined concept features from a contextual representation. Let T be a set of words-in-context, and let H be the set of representations of T, such that $\vec{h_t} \in \mathbb{R}^d$ is the contextualized representation of the word $t \in T$. Let F be a linguistic feature that we hypothesize is encoded in H. Given H and the values f_t of the feature F for each word, INLP returns a set of m linear classifiers, each of which predicts F with above-chance accuracy. Each of these classifiers is a vector in $\mathbb{R}^d$, and corresponds to a direction in the representation space. The m vectors can be arranged in a matrix $\mathbf{W}^{m \times d}$. Since the m classifiers are mutually orthogonal, so are the rows of $\mathbf{W}$. Each linear classifier can be interpreted as defining a *separating plane*, which is intended to partition the space, as much as possible, according to the values of the feature F. In our case, F can take one of two values—whether or not a given word t is in an RC—and each direction in $\mathbf{W}$ is intended to separate words that are inside RCs from words

Abstract structure	Example
Unreduced Object RC (ORC)	The conspiracy that the employee welcomed divided the beautiful country.
Reduced Object RC (ORRC)	The conspiracy the employee welcomed divided the beautiful country.
Unreduced Passive RC (PRC)	The conspiracy that was welcomed by the employee divided the beautiful country.
Reduced Passive RC (PRRC)	The conspiracy welcomed by the employee divided the beautiful country.
Active Subject RC (SRC)	The employee that welcomed the conspiracy quickly searched the buildings.
P/OR(R)C-matched Coordination	The conspiracy welcomed the employee and divided the beautiful country.
SRC-matched Coordination	The employee welcomed the conspiracy and quickly searched the buildings.

Table 1: Examples of sentences generated from the 5 RC structures, the 2 coordination structures. Elements which only occur in a subset of the examples are indicated in grey. This table is copied from Prasad et al. (2019).

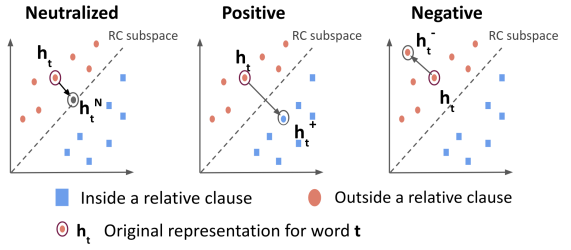


Figure 2: Generating counterfactual representations. A representation $\vec{h}^t$ of a word outside of an RC is transformed to create counterfactual mirror images $\vec{h_t^-}$,$\vec{h_t^+}$ with respect to an empirical RC subspace. The RC subspace here is a 1-dimensional line for illustrative purposes; in practice we use an 8-dimensional subspace.

that are outside them.[2]

The *feature subspace*—the space spanned by all the learned directions ($R = span(\mathbf{W})$)—is a subspace of the original representation space that contains information useful to linearly decode F with high accuracy. The orthogonal complement of R (the *null space*; N) is a subspace in which it is *not* possible to predict F with high accuracy.

3 AlterRep: Generating Counterfactual Representations

The goal of AlterRep is to generate, based on a model's contextual representations of a set of words, a set of counterfactual representations that modify the encoding of a feature F while leaving all other aspects of the representations intact.[3] If swapping these counterfactual representations for the model's original representations changes the model's probability distribution over predicted words in a way that aligns with the feature's linguistic functions, we say that the model uses F for word prediction in a manner that is consistent with the grammar of English.

For our case study, we use a feature with two possible values: '+' if the word is inside an RC and '−' if it is not.[4] We generate two counterfactual representations: h_t^+, which encodes that the word t is inside an RC—regardless of its actual syntactic position in the sentence—and h_t^-, which is similar to h_t^+ in all respects except it encodes that t is not in an RC. Our method allows us to generate h_t^+ and h_t^- irrespective of the feature value encoded in the original representation h_t. If the model uses this feature appropriately, we expect h_t^+ and h_t^- to lead to different predictions in contexts where correct word predictions depend on determining whether or not the word is inside an RC.

Row-space and Null-space Recall that INLP defines a feature space R where the property of interest is encoded, and a complement subspace N where it is not.

We can project any word representation $\vec{h_t}$ to the feature subspace (here, the RC subspace) or to the null space, resulting in the vectors $\vec{h_t^R}$ and $\vec{h_t^N}$, respectively: $\vec{h_t^R}$ maintains the information needed to predict F from $\vec{h_t}$, while $\vec{h_t^N}$ maintains all information which is *not* relevant for predicting F. INLP can be used to generate "amnesic counterfac-

[2] In this paper, we make the simplifying assumption that sentences do not contain RCs that are nested within one another (cf. Lakretz et al. 2021). To accommodate such sentences in future work, an integer feature could be used whose value would be 0 if the word is outside any RC; 1 if it is inside an RC of depth 1; 2 if it is inside an RC of depth 2; and so on. As long as we specify a bound on the embedding depth, this feature would still be categorical, and a variant of our method could still be used.

[3] We aim to propose a concrete instantiation that *approximates* the counterfactual.

[4] In the experiments below, we will only apply this procedure to sets of representations of words that are all in a particular *type* of RC (for example, Object RCs). We do, however, test whether the representations of RC boundary generalize across RC types; see Prediction 3 in §5.

tuals" (Elazar et al., 2021), which do not encode a given property, even if the original representation did encode that property. In the next paragraph we propose a way to use this algorithm to *manipulate* the value of the feature, rather than remove it.

Generating Counterfactual Representations We obtain the counterfactual representations $\vec{h_t^+}$ and $\vec{h_t^-}$ as follows. As we discussed in Section 2.2, INLP identifies planes—one for each direction (row) in $\mathbf{W}$—each of which linearly divides the word-representation space into two parts: words that are in an RC and words that are not. From the representation $\vec{h_t}$ of a word t that is not in an RC, we can generate $\vec{h_t^+}$ by pushing $\vec{h_t}$ across the separating plane towards the representations of words that are inside an RC . Similarly, we can generate $\vec{h_t^-}$ by moving $\vec{h_t}$ *further away* from that plane (see Figure 2).[5]

How do we move the representation of a word away from or towards the separating planes? Recall that the feature subspace R and the nullspace N are orthogonal complements, and consequently any vector $\vec{v} \in \mathbb{R}^n$ can be represented as a sum of its projections on R and N. Further, by definition, the vector's projection on R is the sum of its projections on the RC directions $\vec{w} \in \mathbf{W}$. Thus, we can decompose $\vec{h_t}$ as follows, where $\vec{h_t^w}$ is the orthogonal projection of $\vec{h_t}$ on direction $\vec{w}$:

$$\vec{h_t} = \vec{h_t^N} + \vec{h_t^R} = \vec{h_t^N} + \sum_{\vec{w} \in W} \vec{h_t^w} \quad (1)$$

For any word t, we expect a positive counterfactual $\vec{h_t^+}$ to be classified as being *inside* an RC, with high confidence, according to all *original* RC directions $w \in \mathbf{W}$ — that is, $\forall\, w \in \mathbf{W}, w^T \vec{h_t^+} > 0$. Conversely, we expect a negative counterfactual to be classified as *not* being in an RC, i.e., $\forall\, w \in \mathbf{W}, w^T \vec{h_t^-} < 0$.

To enforce these desiderata, we create positive and negative counterfactuals as follows, where $SIGN(x) = 1$ if $x \geq 0$ and 0 otherwise, and α is a positive scalar hyperparameter that enhances or dampens the effect of the intervention.

$$\vec{h_t^-} = \vec{h_t^N} + \alpha \sum_{\vec{w} \in \mathbf{W}} (-1)^{SIGN(w^T \vec{h_t})} \vec{h_t^w} \quad (2)$$

$$\vec{h_t^+} = \vec{h_t^N} + \alpha \sum_{\vec{w} \in \mathbf{W}} (-1)^{1-SIGN(w^T \vec{h_t})} \vec{h_t^w} \quad (3)$$

In both cases, we *subtract* a direction $\vec{h_t^w}$, flipping its sign, if the sign constraints are violated, that is, if $w^T \vec{h_t} > 0$ for $\vec{h_t^-}$ and if $w^T \vec{h_t} < 0$ for $\vec{h_t^+}$. Geometrically, flipping the sign of a direction $\vec{h_t^w}$ in Equations 2 and 3 is equivalent to taking a *mirror image* with respect to a direction w (Figure 2). This enforces the sign constraints: all classifiers w predict the negative or positive class, respectively (see Appendix §A.1 for a formal proof).

4 Experimental Setup

Our overall goal is to assess the causal effect of RC boundary representations on our models' agreement behavior when subject-verb dependencies span an RC (that is, where an RC intervenes between the head of the subject and the corresponding verb). We test whether we can modify the representation of the masked verb outside the RC such that, compared to the original representations, the model assigns higher probability to either the correct form (after negative intervention) or to the incorrect one (after positive intervention). We first describe the models we use (§4.1), then the dataset we use to obtain RC subspaces and generate counterfactual representations (§4.2), and finally the dataset we use to measure the models' agreement prediction accuracy in sentences containing RCs, before and after the counterfactual intervention (§4.3).

4.1 Models

We use BERT-base (12 layers,768 hidden units) and BERT-large (24 layers, 1024 hidden units) (Devlin et al., 2019), as well as the smaller BERT models released by Turc et al. (2019): BERT-medium (8 layers × 512 hidden units), BERT-small (4 layers, 512 hidden units), BERT-mini (4 × 256), and BERT-tiny (2 × 128). In all experiments, we intervene on a single layer at a time, and continue the forward pass of the original model through the following layers.

4.2 Generating Counterfactual Representations

Datasets To create the training data for the INLP classifiers, we used the templates of Prasad et al.

[5] For a word t that is *inside* an RC, the reverse computations would be required: to generate h_t^+ we would move $\vec{h_t}$ further away from the separating plane, whereas to generate h_t^- we would move $\vec{h_t}$ across the separating plane.

(2019) to generate five lexically matched sets of semantically plausible sentences, one for each type of RC outlined in Table 1, as well as two additional sets of sentences without RCs; these included sentences with nearly the same word order and lexical content as the sentences in the other sets. Each set contained 4800 sentences. All verbs in the training sentences were in the past tense; this ensured that the subspaces we identified did not contain information about overt number agreement, making it unlikely that AlterRep will alter agreement-related information that does not concern RCs.

Identifying and Altering RC Subspaces To identify RC subspaces, we used INLP with SVM classifiers as implemented in scikit-learn. We identified different subspaces for each of the five types of RCs listed in Table 1. For example, in (5), the bolded words were considered to be in the RC.

(5) My cousin **that liked the book** hated movies.

For the negative examples, we took representations of words outside of the RC, either from outside the bolded region of the same sentence, or from inside or outside the bolded region of the coordination control sentence.

(6) My cousin **liked the book** and hated movies.

We selected the negative examples in this manner for two reasons: first, to ensure that the same word served as a positive example in some context and as a negative example in others (e.g., *book* in (5) and (6)); and second, to ensure that the same RC sentence included both positive and negative examples (e.g., *book* and *cousin* in (5)).

Hyperparameters INLP has a hyperparameter m which sets the dimensionality of the RC subspace; this parameter trades off exhaustivity against selectivity.[6] We set $m = 8$; In Appendix §A.3 we demonstrate that the trends we observe are not substantially affected by this parameter.

AlterRep has an hyperparameter, α, that determines the magnitude of the counterfactual intervention (§4.2). We use $\alpha = 4$; In Appendix §A.4 we show that the trends we observe are similar for other values of α.

[6]In particular, running INLP for 768 iterations—the dimensionality of BERT representations—yields the original space, which is exhaustive but not useful in distilling RC information.

4.3 Measuring the Effect of the Intervention on Agreement Accuracy

Dataset We measure the models' agreement prediction accuracy using a subset of the Marvin and Linzen (2018) dataset in which the subject is modified by an RC. The noun inside the RC either matched (7) or mismatched (8) the subject of the matrix clause in number:

(7) The skater **that the officer loves** is/are happy.

(8) The skater **that the officers love** is/are happy.

The Marvin and Linzen dataset contains sentences where the intervening RC is either a subject RC or a (reduced or unreduced) object RC. We augmented this dataset with lexically matched sentences with (reduced or unreduced) passive RC interveners, using attribute varying grammars (Mueller et al., 2020). Finally, we only considered sentences with copular main verbs (*is* and *are*) to ensure that both the singular and plural forms of the verb are highly frequent. We used 1750 sentences per construction.

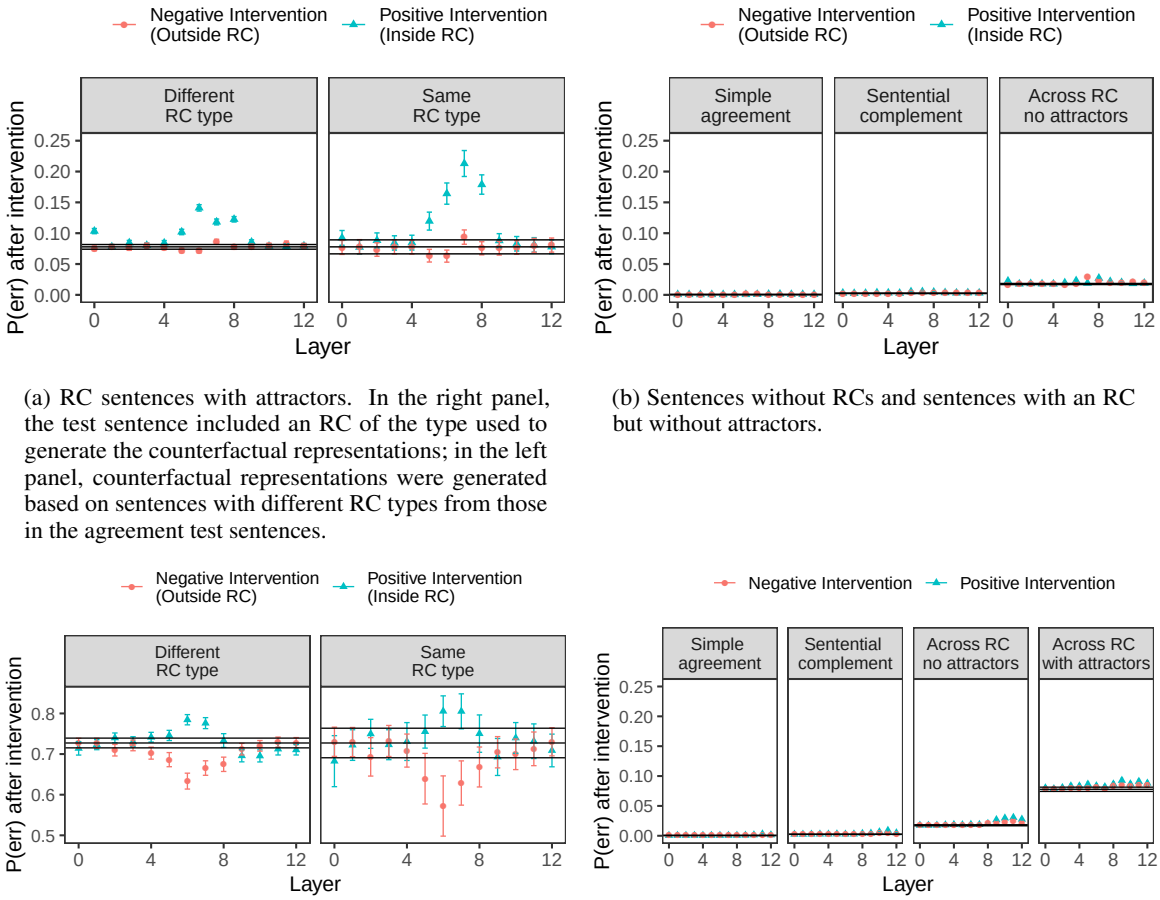
Computing Agreement Accuracy We performed masked language modeling (MLM) on the dataset described earlier in this section. In each sentence, we masked the copula, started the forward pass, performed the intervention on the representation of the masked copula in the layer of interest, and continued with the forward pass to obtain BERT's distribution over the vocabulary for the masked token. We repeated this process for each layer separately. We then computed the probability of error, normalized within the two copulas *is* and *are* (Arehalli and Linzen 2020):

$$P(Err) = \frac{P(Verb_{Incorrect})}{P(Verb_{Incorrect}) + P(Verb_{Correct})} \quad (4)$$

In Appendix §A.5, we present results where the metric of success is accuracy, that is, the percentage of cases where the model assigned a higher probability to the verb with the correct number (Marvin and Linzen, 2018). These results are qualitatively similar.

5 Predictions

As discussed earlier, a system that computed agreement in accordance with the grammar of English would determine the number of the masked verb in a sentence like (9) based on the number of *officers*, because both *officers* and the [MASK] token



(a) RC sentences with attractors. In the right panel, the test sentence included an RC of the type used to generate the counterfactual representations; in the left panel, counterfactual representations were generated based on sentences with different RC types from those in the agreement test sentences.

(b) Sentences without RCs and sentences with an RC but without attractors.

(c) Sentences where before the intervention the model assigned a higher probability to the ungrammatical than the grammatical verb. Note the y-axis differs from other plots (reflecting the higher original error probability).

(d) Intervention from counterfactual representations generated from 10 random subspaces.

Figure 3: Change in probability of error with negative and positive counterfactual BERT-base representations (red circle and cyan triangle respectively). Horizontal lines indicate probability of error with the original representations without any intervention: the middle line is the mean accuracy across all items prior to intervention and the upper and lower lines indicate accuracy two standard errors away from the mean accuracy. Error bars reflect two standard errors from the mean probability of error after intervention.

are outside the RC; the number of the RC-internal noun *skater* should be ignored.

(9) The officers **that love the skater** [MASK] nice.

We can derive the following predictions for applying AlterRep to a system that follows this strategy:

Prediction 1: Impact on Error Probability in RC Sentences with Attractors. In RC sentences where the main clause subject differs in number from the RC subject, error probability will be higher with the counterfactual h^{+}_{MASK}, which encodes (incorrectly) that [MASK] is *inside* the RC, than with the original representation h_{MASK}. Conversely, error probability will be lower with h^{-}_{MASK}, which encodes (correctly) that [MASK] is *outside* the RC, than with the original h_{MASK}.

Prediction 2: No Impact on Other Sentences. We do not expect a difference in error probability between the original and counterfactual representations in all other sentences. This should be the case for sentences with RCs where the nouns inside and outside the RC match in number, as in (10):

(10) The officers **that love the skaters** [MASK] nice.

Since both *officers* and *skaters* are plural, most plausible agreement prediction strategies would make the same predictions regardless of whether [MASK] is analyzed as being inside the RC or out-

side it. Consequently, intervening on the encoding of RC boundaries is not expected to systematically change the model's predictions.

Likewise, since the interventions are designed to modulate the encoding of RC-related properties, we do not expect the interventions to impact number prediction in sentences without RCs such as (11) and (12):[7]

(11) The officer [MASK] nice.

(12) The bankers knew the officer [MASK] nice.

Prediction 3: Generalization Across RC Types. If RC boundaries are represented in an abstract way that is shared across different RC types, then the counterfactual representations will affect error probability in the same way regardless of whether the counterfactual representations were generated from subspaces estimated from sentence with the same RC type as the target sentences, or from sentences with different RC types.

6 Results

Counterfactual Intervention in the Middle Layers of BERT-base Modulates Agreement Error Rate in RC Sentences with Attractors. We begin by discussing experiments where subspaces were estimated from sentences with the same type of RC as the test sentences with agreement; we report results averaged across the five RC types. Interventions using counterfactual representations generated from the middle layers of the BERT-base (5–8 out of 12) resulted in changes in the probability of error which partially aligned with Prediction 1 (Figure 3a). In sentences with attractors, using the positive counterfactual h^{+}_{MASK} resulted in an increase in the probability of error (a maximum increase of 14 percentage points in layer 7). Conversely, using the negative counterfactual h^{-}_{MASK} generated from layers 5 and 6 resulted in a decrease in the probability of error. This decrease was much smaller (a maximum decrease of 2 percentage point in layer 6) and there was overlap in the error bars for the probability of error before and after intervention.

It is likely that the smaller effects of the negative counterfactual intervention are due to the fact that accuracy before the intervention was very high (95%) and the probability of error very low (8%), leaving very little room for change: in most cases, the original representation already correctly encoded the verb is outside the RC. In a follow-up analysis, we only considered sentences in which the model originally assigned a higher probability to the ungrammatical than the grammatical form. In these examples the decrease in probability of error was larger (a maximum decrease of 16 percentage points in layer 6; see Figure 3c).

While only RC interventions in the middle layers elicit the expected behavioral outcomes, probing accuracy for RC information was high for all layers (Appendix §A.2), giving further evidence to the dissociation between correlational and causal methods: probing can identify aspects of the representations that do not affect the model's behavior.

Interventions on RC Boundary Representations Generalize Across RC Types, but not Further. In line with Prediction 3, we observed a qualitatively similar pattern of change in error probabilities even when the counterfactuals were generated from subspaces estimated from a *different* RC type than the RC in the agreement test sentences. The effects were smaller, however. This suggests that while BERT's representation of RC boundaries is partly shared across different RC types, there are also structure-specific RC boundary representations. The effect of the intervention also aligned with Prediction 2: in constructions where we do not expect RC boundaries to affect predictions—sentences without attractors and those without RCs—we did not observe significant changes in error probability (Figure 3b).

Intervention Based on Random Subspaces Does Not Produce Interpretable Results. To tease apart the effect of the RC-targeted intervention from intervening on *any* subspace with the same dimensionality, we generated counterfactual representations from 10 random subspaces and repeated our analysis.[8] While we observed very small changes in probability of error in some cases, the pattern of changes resulting from this intervention did not align with any of our predictions (see Figure 3d). This suggests that the change in probability of error that resulted from intervening with RC subspaces was not merely a by-product of intervening on a large enough subspace of BERT's original representation space.

[7] If models encoded boundaries of all embedded clauses similarly we would expect a change in prediction for (12).

[8] We generated a random subspace by sampling standard Gaussian vectors instead of the INLP matrix $\mathbf{W}$, and then employing the same procedure described in §3.

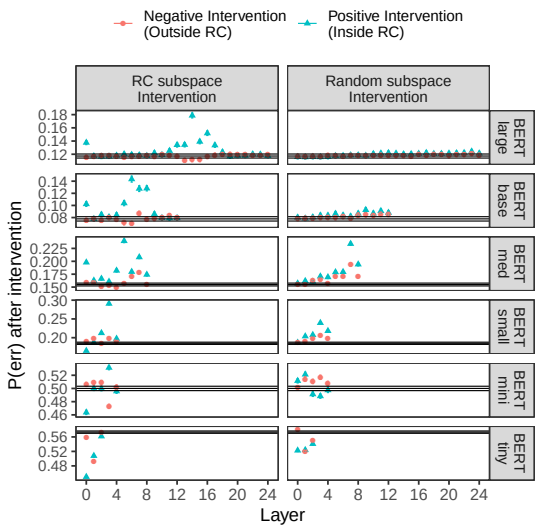


Figure 4: Change caused by counterfactual representations in agreement error probability across RCs with attractors for different BERT variants. Note that the baseline performance prior to intervention (marked by black horizontal lines) is different between models.

Intervening on the Middle Layers of Other BERT Variants Yielded Qualitatively Similar Results. We repeated the experiments on BERT-large and four smaller versions of BERT, trained on the same amount of data as the BERT-base model (Turc et al., 2019). As with BERT-base, intervening on the middle layers of BERT-large (12–17 out of 24) with the RC subspaces—but not the random subspaces—resulted in predicted changes in the probability of error. Compared to BERT-base, the smaller models showed a greater change in the probability of error as a result of intervention with counterfactuals generated from random subspaces. However, when the counterfactual representations were generated from particular layers—4 and 5 (out of 8) in BERT-medium, 3 (out of 4) in BERT-mini and 2 (out of 4) in BERT-small—the change in error probability aligned with Prediction 1 over and above the changes from intervening with random subspaces. In all of these layers, intervening with the positive but not the negative counterfactual resulted in an increase in the probability of error. No such layer was observed for BERT-tiny, which has only 2 layers (see Figure 4).

7 Discussion

We proposed an intervention-based method, AlterRep, to test whether language models use the linguistic information encoded in their representations in a manner that is consistent with the grammar of the language they are trained on. For a given linguistic feature of interest, we generated counterfactual contextual word representations by manipulating the value of the feature in the original representations. Then, by replacing the original representations with these counterfactual variants, we characterized the change in word prediction behaviour. By comparing the resulting change in word prediction with hypotheses from linguistic theory about how specific values of the feature are expected to influence the probabilities over predicted words, we investigated whether the model *uses* the feature as expected.

As a case study, we applied this method to study whether altering the information encoded about RC boundaries in the contextual representations of masked verbs in different BERT variants influences the verb's number inflection in a manner that is consistent with the grammar of English. We found that while all layers of the BERT variants encoded information about RC boundaries, only the information in the middle layers influenced the masked verb's number inflection as predicted by English grammar. We also found that in BERT-base, counterfactual representations based on subspaces that were learned from sentences with one type of RC influenced the number inflection of the masked verb in sentences with other types of RCs; this suggests that the model encodes information about RC boundaries in an abstract manner that generalizes across the different RC types.

Caveat: Linear Analysis of a Non-linear Network AlterRep interventions are based on concept subspaces identified using linear classifiers, but most neural networks components, including BERT layers, are non-linear. It is possible, then, that subsequent non-linear layers transform the counterfactual representation in a way that is not amenable to analysis using our methods. As such, while we can conclude from a positive result that the feature in question causally affects the model's behavior, negative results should be interpreted more cautiously.

Future Work Future work can apply this method to test linguistically motivated hypotheses about a wide range of structures and tasks. For example, linguistic theory predicts that information about semantic roles (like agent and patient) is crucial for tasks such as natural language inference (NLI) that require reasoning about sentence meaning. To test

if NLI models use semantic roles as predicted by linguistic theory, we can use AlterRep to replace the original representations with counterfactual representations where the patient is encoded as the agent (and vice versa), and measure the change in performance on NLI, especially on challenge sets such as HANS (McCoy et al., 2019) that evaluate sensitivity to these properties.

8 Related Work

Probing and Causal Analysis Behavioral tests of neural models, such as the ability of the model to master agreement prediction (Linzen et al., 2016; Gulordava et al., 2018; Goldberg, 2019), have exposed both impressive capabilities and limitations. These paradigms focus on the model's output, and do not link the behavioral output with the information encoded in its representations. Conversely, probing (Adi et al., 2017; Conneau et al., 2018; Hupkes et al., 2018) does not reveal whether the property recovered by the probe affects the original model's prediction in any way (Hewitt and Liang, 2019; Tamkin et al., 2020; Ravichander et al., 2021). This has sparked interest in identifying the *causal* factors that underlie the model's behavior (Vig et al., 2020; Feder et al., 2020; Voita et al., 2020; Kaushik et al., 2020; Slobodkin et al., 2021; Pryzant et al., 2021; Finlayson et al., 2021).

Counterfactuals The relation between counterfactual reasoning and causality is extensively discussed in social science and philosophy literature (Woodward, 2005; Miller, 2018, 2019). Attempts have been made to generate counterfactual examples (Maudslay et al., 2019; Zmigrod et al., 2019; Ross et al., 2020; Kaushik et al., 2020; Hvilshøj et al., 2021) and recently to derive counterfactual representations (Feder et al., 2020; Elazar et al., 2021; Jacovi et al., 2021; Shin et al., 2020; Tucker et al., 2021). Contrary to our approach, previous attempts to generate counterfactual representations were either limited to amnesic operations (i.e., focused on the *removal* of information and not on *modifying* the encoded information) or used gradient-based interventions, which are expressive and powerful, but less controllable. Our linear approach is guided by well-defined desiderata: we want all linear classifiers trained on the original representation to predict a specific class for the counterfactual representations, and we *prove* that is the case in Appendix §A.1.

Representations and Behavior Previous work bridging the gap between representations and behavior includes Giulianelli et al. (2018), who demonstrated that back-propagating an agreement probe into a language model induces behavioral changes and improve predictions. Lakretz et al. (2019) identified individual neurons that causally support agreement prediction. Prasad et al. (2019) used similarity measures between different RC types extracted using behavioural methods to investigate the inner organization of information within the model. Closest to our work is Elazar et al. (2021), where the authors applied INLP to "erase" certain distinctions from the representation, and then measured the effect of the intervention on language modeling. We extend INLP to generate flexible counterfactual representations (§3) and use these to instantiate hypotheses about the linguistic factors that guide the model's behavior.

9 Conclusions

We proposed an intervention-based approach to study whether a model uses a particular linguistic feature as predicted by the grammar of the language it was trained on. To do so, we generated counterfactual representations in which the linguistic property under consideration was altered but all other aspects of the representation remained intact. Then, we replaced the original word representation with the counterfactual one and characterized the change in behaviour. Applying this method to BERT, we found that the model uses information about RC boundaries that is encoded in its word representations when inflecting the number of masked verb in a manner consistent with the grammar of English. We conclude that AlterRep is an effective tool for testing hypotheses about the function of the linguistic information encoded in the internal representations of neural LMs.

Acknowledgements

This work was supported by United States–Israel Binational Science Foundation award 2018284, and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 802774 (iEXTRACT). We thank Robert Frank for fruitful discussion of an early version of this work, and Marius Mosbach for his helpful comments.

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.

Suhas Arehalli and Tal Linzen. 2020. Neural language models capture some, but not all, agreement attraction effects. In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*, pages 370—376.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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. 2020. CausaLM: Causal model explanation through counterfactual language models. *CoRR*, abs/2005.13407.

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.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. Syntaxgym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem H. Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 240–248. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 1195–1205.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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.

Frederik Hvilshøj, Alexandros Iosifidis, and Ira Assent. 2021. ECINN: efficient counterfactuals from invertible neural networks. *CoRR*, abs/2103.13701.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. *CoRR*, abs/2103.01378.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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*, page 104699.

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.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5266–5274. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Tim Miller. 2018. Contrastive explanation: A structural-model approach. *CoRR*, abs/1811.03163.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington. Association for Computational Linguistics.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Reid Pryzant, Dallas Card, Dan Jurafsky, Victor Veitch, and Dhanya Sridhar. 2021. Causal effects of linguistic properties. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4095–4109. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7237–7256. Association for Computational Linguistics.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard H. Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3363–3377. Association for Computational Linguistics.

Alexis Ross, Ana Marasovic, and Matthew E. Peters. 2020. Explaining NLP models via minimal contrastive editing (mice). *CoRR*, abs/2012.13985.

Seungjae Shin, Kyungwoo Song, JoonHo Jang, Hyemi Kim, Weonyoung Joo, and Il-Chul Moon. 2020. Neutralizing gender bias in word embedding with latent disentanglement and counterfactual generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 3126–3140. Association for Computational Linguistics.

Aviv Slobodkin, Leshem Choshen, and Omri Abend. 2021. Mediators in determining what processing BERT performs first. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 86–93. Association for Computational Linguistics.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah D. Goodman. 2020. Investigating transferability in pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1393–1401. Association for Computational Linguistics.

Mycal Tucker, Peng Qian, and Roger Levy. 2021. What if this modified that? syntactic interventions via counterfactual embeddings. *CoRR*, abs/2105.14002.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Sahil Verma, John P. Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *CoRR*, abs/2010.10596.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. 2020. Causal mediation analysis

for interpreting neural NLP: the case of gender bias. *CoRR*, abs/2004.12265.

Elena Voita, Rico Sennrich, and Ivan Titov. 2020. Analyzing the source and target contributions to predictions in neural machine translation. *CoRR*, abs/2010.10907.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

James Woodward. 2005. *Making things happen: A theory of causal explanation*. Oxford university press.

Ran Zmigrod, S. J. Mielke, Hanna M. Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1651–1661. Association for Computational Linguistics.

A Appendix

A.1 Correctness of the Counterfactual Generation

In this appendix, we prove that the method presented in §3 is guaranteed to achieve its goal: the negative counterfactual $\vec{h_t^-}$ would be classified as belonging to the negative class, and the positive counterfactual $\vec{h_t^+}$ would be classified as belonging to the positive class, according to all the linear classifiers trained on the original representation.

We base our derivation on the decomposition presented in §3:

$$\vec{h_t} = \vec{h_t^N} + \vec{h_t^R} = \vec{h_t^N} + \sum_{\vec{w} \in W} \vec{h_t^w} \tag{5}$$

Where N is the nullspace of the INLP matrix $\mathbf{W}$, R is its rowspace, and $\vec{h_t^N}$ and $\vec{h_t^R}$ are the orthogonal projection of a representation $\vec{h_t}$ to those subspaces, respectively.

We focus on the negative counterfactual; The proof for the positive counterfactual is similar. In the proceeding discussion, $w_j \in \mathbb{R}^d$ is an arbitrary linear classifier trained on the jth iteration of INLP (one of the rows in the matrix $\mathbf{W}$). w_j predicts a negative or positive class $\hat{y} \in \{0, 1\}$ according to the decision rule $\hat{y} = SIGN(w_j^T h_t)$ [9]. We denote by $\vec{h_t^w}$ the orthogonal projection of $\vec{h_t}$ on a direction w, given by $(\vec{h_t}^T w)\vec{w}$.

Claim A.1. *For the negative counterfactual defined by* $\vec{h_t^-} = \vec{h_t^N} + \alpha \sum_{i=0}^{m} (-1)^{SIGN(w_i^T \vec{h_t})} \vec{h_t^{w}}_i$, *it holds that* $\vec{h_t^-}$ *would always be classified to the negative class:* $w_j^T h_t^- < 0$ *for every* w_j *in the original INLP matrix* $\mathbf{W}$.

Proof.

$$w_j^T \vec{h_t^-} = w_j^T (\vec{h_t^N} + \alpha \sum_{i=0}^{m} (-1)^{SIGN(w_i^T \vec{h_t})} \vec{h_t^{w}}_i) \tag{6}$$

$$= w_j^T (\alpha \sum_{i=0}^{m} (-1)^{SIGN(w_i^T \vec{h_t})} \vec{h_t^{w}}_i) \tag{7}$$

$$= \alpha w_j^T ((-1)^{SIGN(w_j^T \vec{h_t})} \vec{h_t^{w_j}}) \tag{8}$$

Where the transition from 6 to 7 stems from $\vec{h_t^N}$ being in the nullsapce of $\mathbf{W}$, so $\forall w \in \mathbf{W}$: $w^T \vec{h_t^N} = 0$; and the transition from 7 to 8 stems from the mutual orthogonality of the INLP classifiers (proved in Ravfogel et al. (2020)): since $\forall\ j \neq i, w_i^T w_j = 0$, it holds that $w_j^T \vec{h_t^{w_i}} = w_j^T ((\vec{h_t}^T w_i) w_i) = (\vec{h_t}^T w_i) w_j^T w_i = 0$.

Now, we consider two cases.

- **Case 1:** $w_j^T \vec{h_t} > 0$, that is, the classifier predicted the positive class on the *original* representation. Then, by 8,

$$w_j^T \vec{h_t^-} = \alpha w_j^T ((-1)^{SIGN(w_j^T \vec{h_t})} \vec{h_t^{w_j}}) \tag{9}$$

$$= \alpha w_j^T (-1) h_t^{w_j} \tag{10}$$

$$= -\alpha w_j^T h_t^{w_j} \tag{11}$$

  Since α is a positive scalar and by assumption $w_j^T \vec{h_t} > 0$, it holds that $w_j^T \vec{h_y^-} < 0$.

- **Case 2:** $S w_j^T \vec{h_t} < 0$, that is, the classifier predicted the negative class on the *original* representation. Then, by 8,

$$w_j^T \vec{h_t^-} = \alpha w_j^T ((-1)^{SIGN(w_j^T \vec{h_t})} \vec{h_t^{w_j}}) \tag{12}$$

$$= \alpha w_j^T h_t^{w_j} \tag{13}$$

$$= \alpha w_j^T h_t^{w_j} \tag{14}$$

  Since α is a positive scalar and by assumption $w_j^T \vec{h_t} < 0$, it holds that $w_j^T \vec{h_y^-} < 0$.

We have proved that regardless of the originally predicted label, all INLP classifiers would predict the negative class on the negative counterfactual, which concludes the proof. □

A.2 Probing Accuracy

In this appendix, we provide probing results for the task on which we run INLP: detecting whether representation was taken over a word inside or outside of an RC. As INLP iteratively trains linear probes, this accuracy is equivalent to the accuracy of the first INLP classifier. In all contextualized layers, we observe probing accuracy of over 90% for all RC types (Figure 5). This contrasts with the intervention results in §6. While it is possible to linearly decode the RC boundary in *all* layers, only in the middle layers do we find that this concept *causally* influences the model's behavior. In other words, good probing performance does not indicate main-task relevancy.

[9] For simplicity, we define $SIGN(x) = 1$ if $x \geq 0$ else 0. 0 corresponds to the negative class.

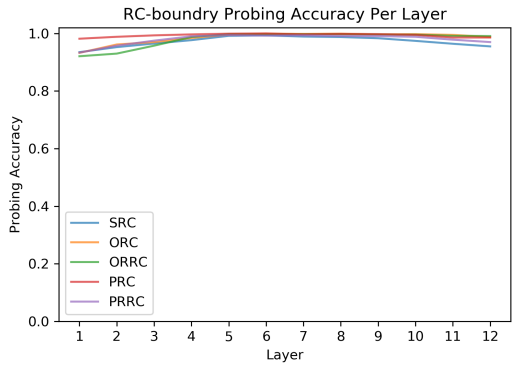
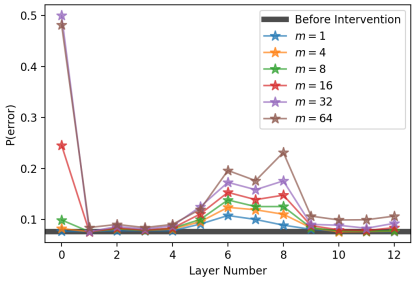
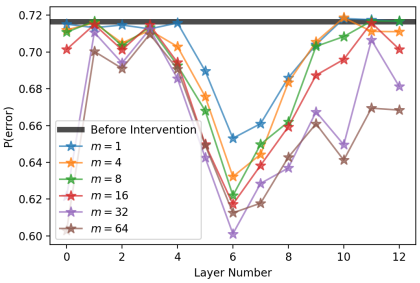


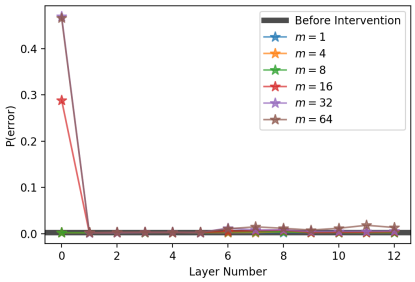
Figure 5: Probing accuracy for the presence of words within or outside of RCs, vs. BERT-base layers, for all the different RC types in our experiments.



(a) **Positive** intervention results on sentences with agreement across RC with attractors.



(b) **Negative** intervention results on sentences with agreement across RC on which the model originally predicted incorrectly.



(c) **Positive** intervention results on sentences with simple agreement and sentences with sentential complements.

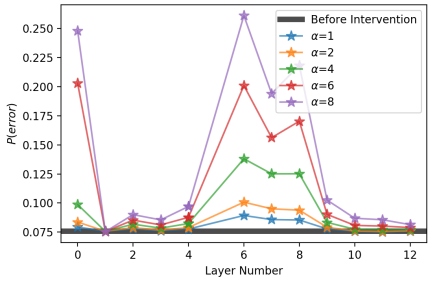
A.3 Influence of the Dimensionality of the RC Subspace

In this appendix, we analyze the influence of the dimensionality m of the RC subspace. Recall that INLP is an iterative algorithm (§2.2). On the ith iteration, the method identifies a single direction $\vec{w_i}$—the parameter vector of a linear classifier—which is predictive of the concept of interest (in our case, RC). The different directions are mutually orthogonal, and after m iterations, the "concept subspace" is the subspace spanned the rows of the matrix $\mathbf{W} = [\vec{w_1}^T, \vec{w_2}^T, \ldots, \vec{w_m}^T]$. In the ith iteration of INLP, the subspace identified so far is *removed* from the representation (by the operation of nullspace projection), and the next classifier $\vec{w_{i+1}}$ is trained to predict the concept over the residual representation. As such, accuracy is expected to decrease with the number of iterations: as the number of iterations increases, the algorithm identifies directions which have a weaker association with the concept. This creates a trade-off between exhaustively – identifying all the directions which are at least somewhat predictive of the concept, and selectivity – identifying only directions which have a meaningful association with the concept.

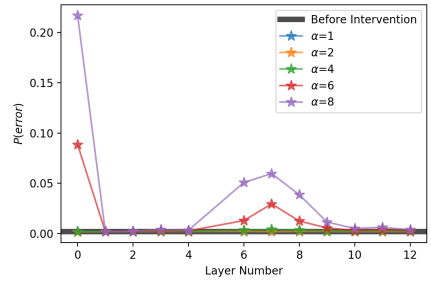
Figure 6a presents positive intervention results for different RC-subspace dimensionality on sentences with agreement across RC with attractors; Figure 6b present negative intervention results on sentences on which the model was originally mistaken. Generally, we observe the same trends under all settings, suggesting our method is relatively robust to the dimensionality of the manipulated subspace. In figure 6c we present the results of intervening on subspaces of different dimensionality, for sentences where we *do not* expect an effect: sentences without attractors, and sentences without RCs. For all contextualized layers we do not see an effect, as expected. For $m = 32$ and $m = 64$, we see an effect on the uncontextualized embedding layer. This effect may hint towards a spurious information encoded in this uncontexualized layer which is used by the model when predicting agreement, but studying this possibility is beyond the scope of this work.

A.4 Influence of α

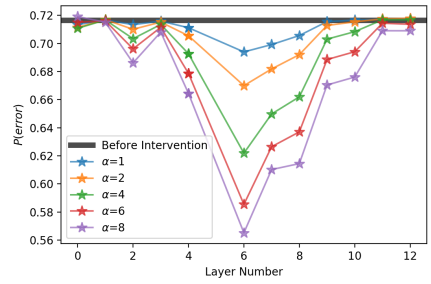
In this appendix, we analyze the influence of the parameter α in the AlterRep algorithm (Section §3) on the BERT-base model. Recall that α dictates the step size one takes when calculating the counterfactual mirror image: $\alpha = 1$ corresponds to exact



(a) **Positive** intervention results on sentences with agreement across RC with attractors.



(b) **Positive** intervention results on sentences with simple agreement and sentences with sentential complements.



(c) **Negative** intervention results on sentences with agreement across RC on which the model originally predicted incorrectly.

Figure 7: Influence of α on the probability of error post intervention.

mirror image, while $\alpha > 1$ over-emphasizes the RC components over which we take the counterfactual mirror image.

In Figures 7a and 7b we focus on the positive intervention, which is expected to increase the probability of error, making the model act as if the masked verb is within the RC; and in Figure 7c we focus on the negative intervention on sentences on which the model was originally mistaken, which is expected to decrease the probability of error.

In Figure 7b we present the results on the control sentences: sentences without agreement across RC. Overall, the trends we observe are similar for different values of α, indicating that AlterRep is relatively robust to the value of this parameter. One exception is the large values of $\alpha = 8$ and to a lesser degree $\alpha = 6$, which result in some increase in the probability of error also in the control sentences, where we do not expect such effect (Figure 7b), albeit this increase is much smaller than the increase on sentences with agreement across RC. With a large-enough α, the new counterfactual representation might diverge too-much from the distribution of the original representations. Notice that when compared with gradient-based methods for generating counterfactuals (Tucker et al., 2021), our linear approach has the advantage of being able to control the magnitude of the intervention with a single controlled parameter, which has a clear geometric interpretation: the extent to which one pushes the representations to one direction or another when taking the mirror image.

A.5 Influence on Accuracy

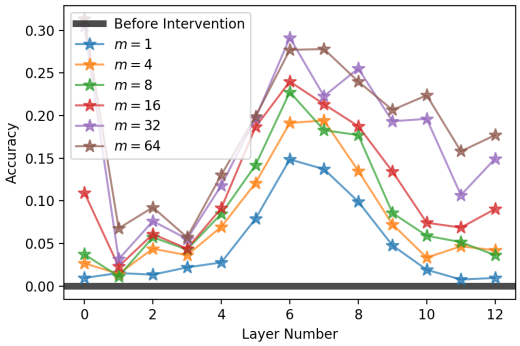


Figure 8: Influence of the *negative intervention* on *accuracy* (the percentage of cases where the model favors the correct form), on sentences on which the model was originally mistaken.

In this appendix, we evaluate the impact of the intervention by its influence on the model's accuracy, calculated as the percentage of cases where the model assigned higher probability to the correct form than to the incorrect form. We focus on the cases on which the model originally predicted incorrectly, Thus, the original accuracy on this group of sentences is 0%. We use a negative intervention, pushing the model to act as if the verb is (correctly) outside of the RC, which is expected to increase its accuracy.

In Section §4.3 we use an alternative measure: probability-of-error. The probability of error is a more sensitive measure, as it might change even when the model's absolute preference for one form

over the other has not. However, it is the absolute ranking which eventually dictates the model's top prediction.

Figure 8 presents the results for different dimensionalities of the RC subspace. The trends are similar to the trends shown by the probability-of-error evaluation measure. Notably, in up to 30% of the cases, it is possible to flip the model's preference from the incorrect to the correct form solely by manipulating a low-dimensional subspace within the 768-dimensional representation space.