ABSTRACTING INFLUENCE PATHS FOR EXPLAINING (CONTEXTUALIZATION OF) BERT MODELS

Kaiji Lu[*], **Zifan Wang, Piotr Mardziel, Anupam Datta**
Electrical and Computer Engineering
Carnegie Mellon University
Mountain View, CA 94089

ABSTRACT

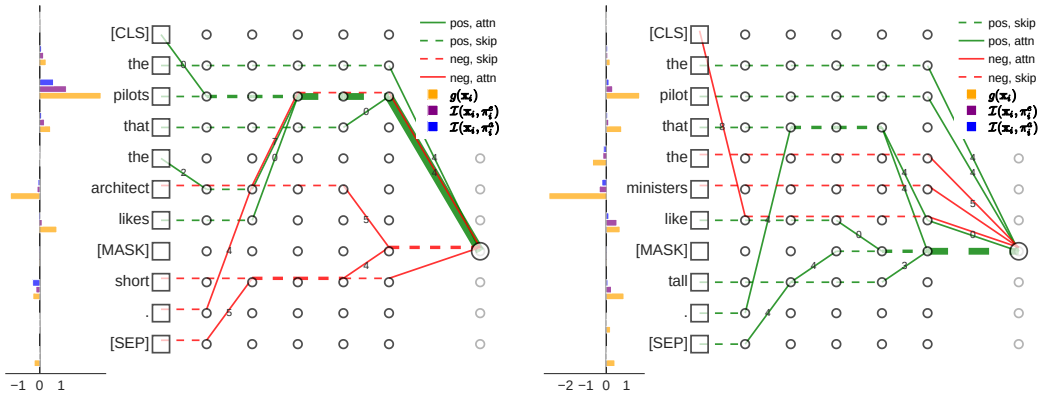
While *"attention is all you need"* may be proving true, we do not yet know *why*: attention-based models such as BERT are superior but how they contextualize information even for simple grammatical rules such as subject-verb number agreement (SVA) is uncertain. We introduce *multi-partite patterns*, abstractions of sets of paths through a neural network model. Patterns quantify and localize the effect of an input concept (e.g., a subject's number) on an output concept (e.g. corresponding verb's number) to paths passing through a sequence of model components, thus surfacing how BERT contextualizes information. We describe guided pattern refinement, an efficient search procedure for finding patterns representative of concept-critical paths. We discover that patterns generate succinct and meaningful explanations for BERT, highlighted by "copy" and "transfer" operations implemented by skip connections and attention heads, respectively. We also show how pattern visualizations help us understand how BERT contextualizes various grammatical concepts, such as SVA across clauses, and why it makes errors in some cases while succeeding in others.

1 INTRODUCTION

The adage *"attention is all you need"* is proving true: attention-based transformer models such as BERT (Devlin et al., 2018) are becoming the state-of-the-art building block for NLP tasks. Beyond computational benefits in parallelization, BERT is responsible for solutions with superior accuracy on most NLP tasks. Like deep models in general, however, BERT is largely inscrutable: we lack understanding of how BERT operates even with respect to simple grammatical rules such as subject-verb number agreement (SVA). On the highest level, BERT has been explained in terms of *contextualization*. Individual word-level information at the input layer is integrated into higher-order structures or concepts in the deeper layers (see the general architecture in Figure 2). While this process has been found to resemble grammatical sentence structures (Hewitt & Manning, 2019; Coenen et al., 2019) in some cases, the principle approaches for discovering contextualization, those based on representations and attention analysis have limitations.

Representation analyses (Lin et al., 2019; Tenney et al., 2019a) demonstrate that relevant linguistic concepts are *associated* with the activations of BERT components (i.e. subject's number associated with the activations of a certain head at a certain layer) but do not tell us how representations come about nor do they localize concepts which we do not apriori relate to the rule (i.e. other representable concepts involved in SVA). Meanwhile, inspection of attention weights as indicators of the flow of information between BERT layers (Clark et al., 2019), requires subjective inference of relevant function (i.e. inference that a certain head may be involved because high attention weights between cells at the subject and cells at the verb). Using attention for explanation has been found to be problematic in other contexts (Brunner et al., 2020; Jain & Wallace, 2019). Analysis of attention further disregards the role of skip connections that do not involve attention at all. Neither approach allows us to track a concept as a causal chain from input to output or to distinguish helpful from hindering representations or flows (hindering information such as contextualization of confounding inputs like unrelated nouns in a sentence lead to errors on SVA).

[*]Correspondence to kaijil@andrew.cmu.edu



(a) plural subject + singular intervening noun(PS) (b) singular subject + plural intervening noun(SP)

Figure 1: Significant patterns π^a extracted by GRP from the attention-level graph for task *SVA across Object Relative Clauses* (Goldberg, 2019; Marvin & Linzen, 2018), in two attractor cases. Left: bar plots of the distributional influence $g(\mathbf{x}_i)$(yellow), $\mathcal{I}(\mathbf{x}_i, \pi_i^e)$ (purple) and $\mathcal{I}(\mathbf{x}_i, \pi_i^a)$ (blue) for each word at position i. Right: significant patterns π_i^a from each input word at position i to quantity of interest (verb number correctness). The square nodes denote the input embeddings and circles denote internal embeddings. Broken lines correspond to skip connection in the attention block while solid lines correspond to connection through (any) attention heads. Attention connections with high traffic are marked with the corresponding attention head number. Line colors represent the sign of influence (red as negative and green as positive).

To address these limitations we introduce *multi-partite*[1] *patterns*, abstractions of sets of paths through a neural model (a graph). Patterns quantify and localize the *effect* of an input concept (e.g. a subject's number) on an output concept (e.g. corresponding verb's number) to *a collection of* paths passing through a sequence of model nodes and/or edges. We describe *guided pattern refinement*, a search procedure for finding patterns representative of concept-critical paths that let us selectively explore the importance of chosen aspects of a model (e.g. in BERT, we can refine patterns showing criticality of certain heads to pathways also showing whether this is due to skip connections or due to attention). To demonstrate the contextualization process, we further extend the experimental framework to integrate impacts of multiple words towards a given concept (as opposed to impact of a single word, e.g. subject on SVA). Example visualization of patterns and their influence for localizing SVA in two sentences are shown in Fig. 1.

Contributions: 1) We describe *multi-partite patterns* for explaining the model-wide contextualization in neural models and *guided pattern refinement* (GPR) to discover influential patterns focusing on model elements of interest. 2) We visualize BERT's contextualization of a variety of grammatical concepts such as subject-verb agreements(SVA) and reflexive anaphora(RA), and show how BERT encode these concepts using correct or incorrect cues. 3) We validate the derived patterns significance towards several concepts by way of concentration and model compression experiments.

We begin with a summary of requisite techniques in Sec. 2. We describe the core elements of our methodology in Sec. 3 and exemplify them for understanding BERT in Sec. 4. We elaborate on related works in Sec. 5 and conclude in Sec. 6.

2 Background

BERT. BERT (Devlin et al., 2018) is a pretrained Transformer (Vaswani et al., 2017) model that has spurred the recent success of NLP. A pretrained BERT model is trained using a Masked Language Modeling (MLM) paradigm where the model learns to guess a masked word in a sentence. The masked word is denoted with the special token `[MASK]`. MLM has been used to evaluate whether

[1] *Multi-partite* because patterns abstract sets of paths in neural models viewed as multi-partite graphs.

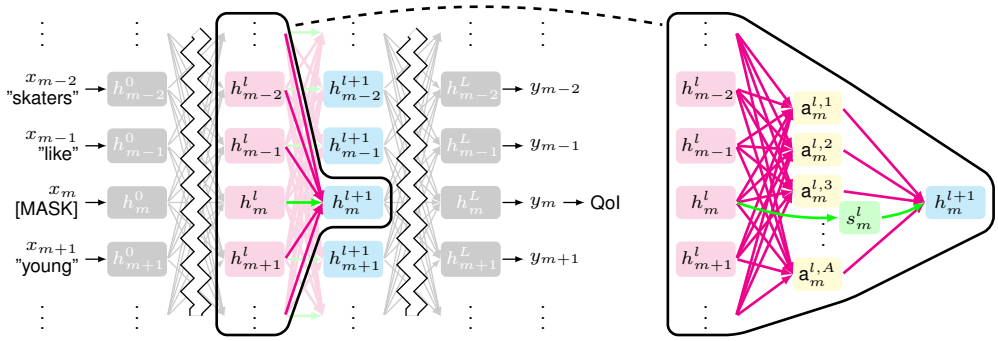


Figure 2: BERT Transformer architecture (left) and details of a transformer layer (right).

a linguistic concept such as SVA is learned by BERT (Goldberg, 2019), by measuring whether BERT assigns a higher probability for the correct verb(e.g. *are* for Figure 1) than the incorrect verb (e.g. *is* for Figure 1) in the `[MASK]` position.BERT first creates embeddings for each input tokens leveraging its positional and segment information. The input embeddings are then passed into a stack of Transformer encoder layers, illustrated in Fig. 2(left). We focus on the attention block in Fig. 2(right), where BERT learns to reweight and combine embeddings from previous layer through multi-head attentions. The embeddings are also "copied" through skip connection, and then combined with attention outputs to produce new embeddings.

Notation. For a BERT model, we use L as the number of Transformer encoder layers, H as the hidden dimension of embeddings at each layer, and A as the number of attention heads. Consider a list of input word embeddings $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_1, ..., \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^d$ as the input for BERT, we denote the output of the l-th layer as $\mathbf{h}^l_{0:N-1}$. First layer inputs are $\mathbf{h}^0_{1:N} \stackrel{\text{def}}{=} \mathbf{x}_{1:N}$. Scores are $\mathbf{y}_i \stackrel{\text{def}}{=} \texttt{softmax}(W\mathbf{h}^L_i), W \in \mathbb{R}^{C\times H}$, where C is the vocabulary size, as the final classification layer and outputs are the predicted word $\hat{y}_i \stackrel{\text{def}}{=} \arg\max_c \mathbf{y}_{i,c}$ for the i-th token. We denote the index of `[MASK]` as m.

Distributional Influence attributes to each DNN input a measure of impact on model output. Saliency (Baehrens et al., 2010) is a simple example that defines influence as the gradient of output with respect to input. Influence can quantify a concept (e.g. SVA) by instrumenting a model's inputs with a *distribution of interest* (DoI) and the output with a *quantity of interest* (QoI) (Leino et al., 2018).

Definition 1 (Distributional Influence) *Given a model $f : \mathbb{R}^d \to \mathbb{R}^n$, an input $\mathbf{x}$, a DoI $\mathcal{D}(\mathbf{x})$ parameterized by an input, and a QoI $q : \mathbb{R}^n \to \mathbb{R}$,* Distributional Influence $g_q(\mathbf{x})$ *quantifies the impact of an input concept defined by the DoI on the output concept defined by the QoI:*

$$g_q(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} \frac{\partial q(f(\mathbf{z}))}{\partial \mathbf{z}}$$

Instantiations defining SVA and RA concepts in BERT models are found in Sec. 4. We use expectation notation for convenience of generality though in our use cases DoI are distributed along a one dimensional path $c : [0, 1] \to \mathbb{R}^d$. Influence is then the line integral $\int_0^1 \frac{\partial q(f(\mathbf{x}))}{\partial \mathbf{x}} (c(\alpha)) \cdot p(\alpha)\, d\alpha$ where p is the p.d.f. of the DoI (Sundararajan et al., 2017).

Influence Paths localize an influence measurement to paths in a neural model, treated as a graph the gradient flows through each node and connections in the graph (Lu et al., 2020). A computation graph $\mathcal{G} \stackrel{\text{def}}{=} (\mathcal{V}, \mathcal{F}, \mathcal{E})$ is a set of nodes, activation functions, and edges, respectively. We assume the graph is directed, acyclic, and does not contain more than one edge per adjacent pair of nodes[2]. A path p in G is a sequence of graph-adjacent nodes $[p_1, p_2, \cdots, p_{-1}]$. We denote the Jacobian of the output of node n_i w.r.t the output of connected (not necessarily directly) predecessor node n_j

[2]The single edge restriction is for notational conveniences to follow; if a given neural model does have more than one edge between adjacent nodes, we can replace duplicate edges with 2-length paths through dummy identity nodes to satisfy this requirement without affecting its semantics.

evaluated at $\mathbf{x}$ as $\partial n_j(\mathbf{x})/\partial n_i(\mathbf{x})$ We write $\nabla_{\mathbf{x}} p$ as the component of the Jacobian passing through path p evaluated at input $\mathbf{x}$ as per chain rule: $\nabla_{\mathbf{x}} p \stackrel{\text{def}}{=} \prod_{i=1}^{-1} \partial p_i(\mathbf{x})/\partial p_{i-1}(\mathbf{x})$.

Definition 2 (Individual Path Influence) *Given a path p of a computation graph $\mathcal{G}$, the* individual path influence *for an input $\mathbf{x}$, or $\chi(\mathbf{x}, p)$ is: $\chi(\mathbf{x}, p) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} [\nabla_{\mathbf{z}} p]$.*

Lu et al. (2020) uses individual path influence to decompose distributional influence to paths and explain internal LSTM behaviour under SVA via the most influential path $\arg\max_{p\in\mathcal{P}} \chi(\mathbf{x}, p)$ where $\mathcal{P}$ are all paths from input to a particular output (normally a QoI).

3 Path Abstraction

Applying individual influence paths to transformer-based models like BERT has computational and conceptual problems. BERT is denser in terms of model connections: each node at each layer integrates information from *all* nodes of the prior layer (as opposed to the pair of short-term and long-term connections in LSTM). This results in an intractable number of influence paths to enumerate, even for processing the simplest of BERT variants. In addition, understanding contextualization requires explaining the impact of all words, such as the role of "that" in Figure 1 as opposed to a single word.

Our approach is three-fold: first we employ abstractions of sets of paths as the localization and influence quantification instrument; second, we discover influential patterns using a greedy search procedure that refines abstract patterns into more concrete ones, keeping the influence high; and third, we consider the collection of influence patterns from every word in a sentence to the quantity of interest.

Definition 3 (Multi-partite pattern) *A* multi-partite pattern *π is a sequence of nodes $[\pi_1, \pi_2, \cdots, \pi_{-1}]$ such that for any pair of nodes π_i, π_{i+1} adjacent in the sequence (not necessarily adjacent in the graph), there exists a path from π_i and π_{i+1}.*

A pattern π abstracts a set of paths, written $\gamma(\pi)$ that follow the given sequence of nodes but are free to traverse the graph between those nodes in any way. Interpreting paths and patterns as sets, we define $\gamma(\pi) \stackrel{\text{def}}{=} \{p \subseteq \mathcal{P} : \pi \subseteq p\}$ where $\mathcal{P}$ is the set of all paths from π_1 to π_{-1}. If every sequence-adjacent pair of nodes is directly connected then the pattern abstracts a single path.

Definition 4 (Pattern influence) *Given a computation graph and a DoI $\mathcal{D}$, the* influence *of a multi-partite influence pattern π, written $\mathcal{I}(\mathbf{x}, \pi)$ is the total influence of all the paths abstracted by the pattern:*

$$\mathcal{I}(\mathbf{x}, \pi) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} \prod_{i=1}^{-1} \frac{\partial \pi_i(\mathbf{x})}{\partial \pi_{i-1}(\mathbf{x})} = \sum_{p\in\gamma(\pi)} \chi(\mathbf{x}, p)$$

The definition indicates that the total influence of paths represented by a pattern can be computed without an enumeration of all those paths. Also note that influence of individual paths may be positive or negative so cancellation in the influence of a pattern which aggregates paths is possible.

Computation Graphs for BERT A given DNN can be expressed by many computational graphs. For computational and interpretability reasons, an ideal graph would contain as few nodes and edges as possible while exposing structures of interest. For BERT in particular, we propose *embedding-level graph $\mathcal{G}_e$* corresponding to the nodes and edges shown in Fig. 2 (left) to explain how the influence of input embeddings flow from one Transformer layer to another and to the eventual prediction of `[MASK]`; and *attention-level graph $\mathcal{G}_a \supset \mathcal{G}_e$* that additionally includes the head nodes as in Fig. 2 (right), a finer decomposition to demonstrate how influence from the input embedding flows through the attention block within each layer. We omit details of how the semantics of BERT can be described by the two graphs.

Note that since the attention level graph contains a superset of the nodes of the embedding level graph, we can interpret embedding level patterns as abstracting paths in both the embedding level graph and the attention level graph. Furthermore, a concrete path in $\mathcal{G}_e$ is a pattern in $\mathcal{G}_a$ as it contains $\mathcal{G}_a$-non-adjacent nodes and thus abstracting multiple paths in $\mathcal{G}_a$. For a given pattern π of

$\mathcal{G}_e$ we can thus write $\gamma_a(\pi)$ as the set of paths it abstracts in $\mathcal{G}_a$ with:

$$\gamma_a(\pi) \stackrel{\text{def}}{=} \bigcup_{p \in \gamma_e(\pi)} \gamma_a(p)$$

Guided Pattern Refinement(GPR) Instead of enumerating the path space of $\mathcal{P}$ for discovering influential paths, we approximate a search by greedily refining patterns while maximizing their influence.

Starting with sources and target nodes s and t along with a pattern $\pi^0 = \{s, t\}$ representing all paths between s and t, we construct π^1 by adding a node from a *guiding set* E^0 that maximizes the influence of the resulting pattern. At the first iteration and subsequently, the guiding set defines a cut of the graph between two sequence adjacent nodes (initially just s and t). The procedure is repeated with additional refinement. At iteration $i+1$, a guiding set E^i defines a cut between nodes s^i and t^i while the cut node that refines the pattern to maximal influence is selected:

$$\pi^{i+1} \stackrel{\text{def}}{=} \pi^i[s^i, t^i \setminus s^i, e^i, t^i]$$
$$e^i \stackrel{\text{def}}{=} \arg\max_{e^i \in E^i} \mathcal{I}(\mathbf{x}, \pi^i\left[s^i, t^i \setminus s^i, e^i, t^i\right])$$

Above, $\pi\left[a, c \setminus a, b, c\right]$ denotes the pattern π in which sequence adjacent nodes a, c are replaced with a, b, c, in their position in the sequence.

Repeating the procedure for some number of steps or until some stopping criterion is reached produces a sequence of patterns with decreasing abstraction: $\gamma(\pi_{i+1}) \subseteq \gamma(\pi_i)$. Once a pattern is produced that abstracts a single path, no more refinement can be done though it might not be desirable to continue refinement until that point for interpretability reasons. Also, the choice of guiding sets E^i at each iteration has a big impact on the resulting patterns both in terms of their influence significance and computational requirements of iteration. Smaller sets require fewer options to enumerate but are likely to lead to less influential patterns.

In our experiments we employ a layer-ordered strategy for the embedding-level pattern refinement and then refine the resulting pattern in the attention-level graph. In the embedding-level analysis, at iteration i, we focus on layer i. The guiding set E_i is the cut:

$$\text{(embedding-level guiding set)} \quad E_i \stackrel{\text{def}}{=} \left\{h^l_j\right\}_j$$

The refinement thus proceeds for L iterations (the input layer can be skipped). If the input node is denoted x and the quantity of interest is denoted as q, the refinement process results in a pattern $\pi^e = \left\{x, h^1_{j_1}, h^2_{j_2}, \cdots, h^L_{j_L}, q\right\}$ where j_i are indices designating which embeddings at each level i the abstracted paths traverse.

The attention-level refinement starts with the embedding-level pattern and exposes the attention heads to cut the flow of influence in that starting pattern, also in order of the layers. At iteration i, the cut E_i is:

$$\text{(attention-level guiding set)} \quad E_i \stackrel{\text{def}}{=} \left\{a^{i,k}_{j_i}\right\}_k \cup \left\{s^i_{j_i}\right\}$$

That is, the cut separates embedding nodes $h^i_{j_i}$ and $h^{i+1}_{j_{i+1}}$ with the attention heads $\left\{a^{i,k}_{j_i}\right\}_k$ and a skip edge (modeled as a node) $s^i_{j_i}$. As the attention-level analysis refines the embedding-level analysis, the produced attention-level pattern π^a abstracts a strict subset of the paths of the attention-level graph than the embedding-level pattern π^a. That is, $\pi^e \subseteq \pi^a$ and therefore $\gamma_a(\pi^e) \supseteq \gamma_a(\pi^a)$.

In our experiments, we perform GPR independently for each word, and expand with most positively influential cut node for positively influential words ($g_q(\mathbf{x}_i) > 0$)) , and vice versa with most negatively influential cut node for negative $g_q(\mathbf{x}_i)$. In the following section, we use π_i as the extracted patterns for individual word i, π as a combination of patterns from all words, and π_+ as the combination of patterns for all positively influential words.

4 EVALUATION

We demonstrate GPR by abstracting patterns to significant paths in the embedding and attention-level graphs of BERT. We first discuss the selected linguistic tasks, datasets, model and hyper-

Task	$\lg \lvert\mathcal{P}^e\rvert$	C^e_+	C^e_-	$\lvert\gamma_e(\pi^e_+)\rvert/\lvert\mathcal{P}^e\rvert$	acc.(π^e_+)	acc.(ori.)	acc.(rand.)
SVA							
Obj.	14.4	0.34	0.29	0.06	0.99	0.96	0.50
Subj.	14.4	0.30	0.31	0.09	0.74	1.00	0.50
WSC	13.8	0.27	0.38	0.03	1.00	1.00	0.52
APP	14.4	0.33	0.31	0.06	0.92	1.00	0.55
RA							
NA	14.4	0.31	0.32	0.04	0.68	0.83	0.54
GA	15.4	0.22	0.23	0.02	0.76	0.73	0.65

Table 1: Concentration and Model Compression Results for Embedding-level Patterns π^e. C^e_+ & C^e_-: The percentage of positive/negative embedding-level pattern influence over positive/negative input influence. $\mathcal{P}^e$: a set of all paths (reachable by GPR) from input to QoI. π^e_+: embedding-level patterns for positively influential words. acc.(ori.) and acc.(rand.) are the accuracies of the original model and a randomly compressed model which retains the same number of nodes as π^e_+.

Task	$\lg \lvert\mathcal{P}^a\rvert$	C^a_+	C^a_-	$\lvert\gamma_a(\pi^a_+)\rvert/\lvert\mathcal{P}^a\rvert$	acc.(π^a_+)	acc.(rand.)
SVA						
Obj.	27.6	0.22	0.16	2.0e-7	0.74	0.51
Subj.	27.6	0.18	0.17	2.0e-7	0.56	0.49
WSC	27.0	0.16	0.24	1.8e-7	0.69	0.49
APP	27.6	0.18	0.15	1.6e-7	0.62	0.60
RA						
NA	27.6	0.18	0.18	1.4e-7	0.49	0.52
GA	28.6	0.14	0.15	1.4e-7	0.72	0.55

Table 2: Similar results of concentration and model compression as in Table 1, using attention-level patterns π^a.

parameters, and then discuss quantitative evaluation of GPR in 4.2, and finally demonstrate how to use GPR to explain the contextualization through a case study of SVA in 4.3.

Tasks. We evaluate two linguistic tasks: subject-word agreement (SVA) and reflexive anaphora (RA). We explore different forms of sentence stimuli in each task: object relative clause (Obj.), subject relative clause (Subj.), within sentence complement (WSC), and across prepositional phrase (APP) in SVA; number agreement (NA) and gender agreement (GA) in RA. SVA and RA datasets (Marvin & Linzen, 2018; Lin et al., 2019) are evaluated with MLM in a same way as Goldberg (2019). We sample 200 sentences evenly distributed across different sentence types (e.g. singular/plural subject & singular/plural intervening noun) with a fixed sentence structure from each task, so that the sentence length and the word types in each position are consistent across samples. Examples of each task are found in Appendix A.

QoI and Distributional Influence. We use the same QoI from Lu et al. (2020) where $q(\mathbf{y}_m) \stackrel{\text{def}}{=} y_{m,correct} - y_{m,wrong}$, e.g. $y_{m,\texttt{IS}} - y_{m,\texttt{ARE}}$ for `she [MASK] happy`. We select $\mathcal{D}$ as an uniform distribution over a linear path from $\mathbf{x}_b$ to $\mathbf{x}$ in the input space for each word where the baseline $\mathbf{x}_b$ is defined as the the input embedding of `[MASK]`, as it can be seen as a word with "zero information".

Model. We choose $\text{BERT}_{\text{SMALL}}(L = 6, A = 8)$ from Turc et al. (2019), with comparable performance to $\text{BERT}_{\text{BASE}}$ (See Appendix A). We choose the smaller model also for its ease of visualization and faster convergence for approximating the influence. The analysis for approximating the influence can be found in Appendix. B.1.

4.1 Quantitative Analysis

We adapt two analysis, *concentration* and *model compression* of Lu et al. (2020), to evaluate whether π^e_i and π^a_i are significant (influential) enough to transmit the linguistic concepts from input to output.

Concentration. Given a sentence $\mathbf{x}$, the positive concentration C_+ is the percentage of the total positive π_i^e and π_i^a, over the total positive influence $g(x_i)$, and similarly C_- for negative . High concentration suggests that most of the distributional influence flows across significant patterns, instead of sparely distributed over a large number of patterns (e.g. the number of total paths in an embedding-level graph is more than 2^{14} for a sentence with 11 tokens).

Model Compression. To validate that the extracted patterns are actually significant for transmitting the information from the input embedding to the output QoI, we employ a model compression algorithm inspired by Lu et al. (2020). Since there is no direct way to only retain the computations of extracted patterns in a forward pass, we propose an approximating model compression procedure for both π^e and π^a compression: for each example, we only retain the nodes where significant positive patterns (π_+^e) pass through while "churning" all other nodes. The "churning" and "retaining" operations are conducted in the layer embeddings (or attention heads/skip connections for attention-level graph). In each layer, the embedding nodes (or the attention/skip connection nodes) to be churned will be replaced by the embedding of `[MASK]` (zeros for attention/skip nodes), while the nodes to be retained will remain untouched. The retained and churned node together will be forwardly passed to the next layer using the original model parameters until a new set of nodes needs to be retained or churned.

Results. We demonstrate concentration statistics and model compression using π_i^e and π_i^a, in Table 1 and 2, respectively. In Table 1, we observe that $\mathcal{I}(\mathbf{x}_i, \pi_i^e)$ accounts for a large portion of both positive and negative influence, range from 0.22 to 0.34 and 0.23 to 0.38, respectively, across all tasks. We also discover that the model compressed with π_+^e retains high accuracy, despite only using a really small number of patterns compared to $|\mathcal{P}^e|$. The compression accuracy can also be compared against a random baseline model, which compresses down by retaining the same number of nodes as π_+^e and shows a low binary accuracy around half, indicating that randomly chosen patterns of the same size cannot transmit the signals required for the task.

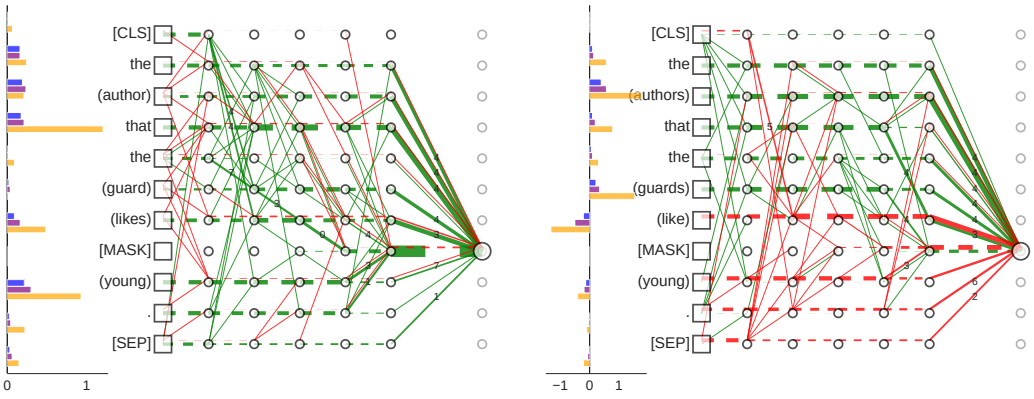
Similar conclusion can be made for attention level patterns π^a in Table 2. Comparing C_+^a with C_+^e $\mathcal{I}(\mathbf{x}_i, \pi_i^a)$ accounts for an even larger portion of $\mathcal{I}(\mathbf{x}_i, \pi_i^e)$, compared to $\mathcal{I}(\mathbf{x}_i, \pi_i^e)$ with respect to the total influence $g(\mathbf{x}_i)$, suggesting influence flow is highly concentrated to either one attention head or "no attention head" (skip connection). The compression accuracy is compromised more as expected due to more nodes churned and replaced, while still more often doing much better than the random baseline.

4.2 Explaining Contextualization of SVA Across Object Relative Clause

In this section, we explain internal contextualization between word embeddings by visualizing the significant patterns π^e and π^a found by GPR. We show results on other tasks in the Appendix. B.2.

Observations of Fig. 3. First we observe that in all four sentence types(PP, PS, SP, SS) of Figure 1 and 3, both words in the subject phrase ("the" and the "noun") exert a positive input influence on the correct prediction of the verb, and the intervening noun exerts positive input influence when agreeing with the subject in number and negative otherwise. This is true for both $\mathcal{I}(\mathbf{x}_i, \pi_i^e)$ and $\mathcal{I}(\mathbf{x}_i, \pi_i^a)$.

"Copy and Transfer" We observe that there are many horizontal lines in the figures, indicating influence travels through layers at the same word position, mostly using skip connections.**We speculate that the reason that attentions can be effectively pruned without compromising the performance in prior works (Michel et al., 2019), is because between most layers, the information from layer embeddings does not travel through attention block at all: they are simply "copied" to the next layer through the skip connections.** Zooming in on π_i^e and π_i^a, we observe that the subject phrase travels through skip connections across the lower layers, and only through attention head 5 in the last layer. This "copy and transfer" procedure indicates that BERT mostly picks up the signal from the subject input embedding without much contextualization, however, exactly how it overcomes(or not) the comparable signals from the attractors is explained the next section. In Appendix B.2 we observe that all the above conclusions are also prevalent in other tasks (such as the contextualization of propositions in WPP task), though different heads might be used for the "transfer" operations in different tasks.



(a) singular subject + singular intervening noun(SS) (b) plural subject + plural intervening noun(PP)

Figure 3: Aggregated patterns across samples of task *SVA across Obj.*. The legend and details for plots can be found in Figure 1. The words within parenthesis represent one instance of the word in that position. For aggregated plots of sentence types in Figure 1, see Appendix B.2.

The Role of "that" Comparing the four difference sentence types, we do observe that the influence from the singular subjects(SS and SP) is weaker than that of the plural subjects, especially compared to that from attractors. The key difference is that "that" behaves differently for singular subjects from plural subjects. In singular subjects, "that" behaves as a singular noun, traveling through the same straightforward pattern as the subject (skip connections + attention head 5); "that" after plural subjects, however, behaves more like a grammatical marker(relativizer): in PP(Figure 3b) and PS(Figure 1a), the patterns from "that" have a strong contextualization edge from itself to either the subject or the clausal verb in the second to last layer. We speculate "that" in plural subject sentences encodes the clausal boundary, therefore enabling the model to identify the main subject and ignore the intervening noun. As a result, attractors in PS have almost no influence, compared to the high negative influence from attractors in SP (a similar phenomena is observed in PP and SS cases). This discrepancy in the behavior of "that" also corroborates lower SVA accuracy in SP case than in PS case (See Appendix A).

5 Related Work

Previous work has shown the encoding of linguistic knowledge within BERT(Devlin et al., 2018). Diagnostic classifiers trained on output and internal representations discover that BERT encodes many types of linguistic knowledge(Elazar et al., 2020; Hewitt & Liang, 2019; Tenney et al., 2019a;b; Jawahar et al., 2019; Klafka & Ettinger, 2020; Liu et al., 2019; Lin et al., 2019), ranging from syntactic concepts to more complicated semantic ones. Goldberg (2019) discover that SVA and RA in complex clausal structures is better represented in BERT compared to an RNN model. This is partially explained by (Coenen et al., 2019; Hewitt & Manning, 2019) which show that contextual embeddings in BERT can encode syntactic structures hierarchically comparable to those represented in a dependency tree. However all these analyses are done on frozen contextual embedding layers; the exact mechanism a concept is encoded from input to output is not explored.

Another line of work in interpreting BERT concerns analyzing the self-attention weights of BERT(Clark et al., 2019; Vig & Belinkov, 2019; Lin et al., 2019), where attention heads are found to have direct correspondences with specific dependency relations. However, attention weights as interpretation devices has been controversial(Serrano & Smith, 2019; Brunner et al., 2020), and empirical analysis has shown that attention can be perturbed or pruned while having the same or even better performance(Kovaleva et al.; Michel et al., 2019; Voita et al., 2019). More importantly, our work demonstrate that attention mechanisms are only part of BERT computation graph, with each attention block complemented by additional architecture such as dense layer and skip connections. The strong influence passing through skip connections also corroborates the findings of Brunner et al. (2020) which find input tokens mostly retain their identity.

Besides pruning attentions, other works(Prasanna et al., 2020; SANH et al.; Jiao et al., 2019) also show that BERT is overparametrized and can be greatly compressed. Our work to some extent corroborates that point by pointing to the sparse gradient flow, while employing model compression only to verify the significance of the extracted patterns.

Recent work introducing influence paths (Lu et al., 2020) offers another form of explanation. Influence paths quantify the effect of model inputs on model outputs that traverse *a particular* path through a model. The approach refines the influence-directed explanations framework of Leino et al. (2018) which instruments a neural model with input a distribution of interest (e.g. datasets meant to exercise a concept such as SVA) and an output quantity of interest (e.g. a measure of SVA) in order to compute a gradient-based measure of input influence (referred to as attribution) such as Integrated Gradient (Sundararajan et al., 2017) and to discover internal model elements most responsible (having highest influence) on the given concept. Lu et al. (2020) decomposed the attribution to path-specific quantities localizing the implementation of the given concept to paths through a model. The authors demonstrated that for LSTM models, a single path is responsible for most of the input-output effect defining SVA, and explored the effects of unhelpful nouns which showed negative influence on SVA. We describe the limitations of this methodology when applied to BERT in Sec. 3.

6 Conclusion

We have demonstrated how to use multi-partite influence patterns to localize a DNN model's handling of a concept of interest and along with a pattern refinement method we how BERT handles subject-verb number agreement and reflexive anaphora. We quantitatively validated the use of influence patterns in BERT by way of compression experiments and the influence concentration of discovered patterns. We qualitatively and visually demonstrated BERT's contextualization in the two tasks using our methodology. Our formalism and methods are general enough to apply to the analysis of other aspects of BERT and other models.

Acknowledgments

This work was developed with the support of NSF grant CNS-1704845 as well as by DARPA and the Air Force Research Laboratory under agreement number FA8750-15-2-0277. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of DARPA, the Air Force Research Laboratory, the National Science Foundation, or the U.S. Government.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. When bert forgets how to pos: Amnesic probing of linguistic properties and mlm predictions. *arXiv preprint arXiv:2006.00995*, 2020.

Yoav Goldberg. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*, 2019.

John Hewitt and Percy Liang. 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)*, pp. 2733–2743, 2019.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. 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)*, pp. 4129–4138, 2019.

Sarthak Jain and Byron C Wallace. Attention is not explanation. 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)*, pp. 3543–3556, 2019.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? 2019.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

Josef Klafka and Allyson Ettinger. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. *arXiv preprint arXiv:2005.01810*, 2020.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. Association for Computational Linguistics.

Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pp. 1–8. IEEE, 2018.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside bert's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253, 2019.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

Kaiji Lu, Piotr Mardziel, Klas Leino, Matt Fredrikson, and Anupam Datta. Influence paths for characterizing subject-verb number agreement in LSTM language models. Association for Computational Linguistics, 2020.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2018.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pp. 14014–14024, 2019.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*, 2020.

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF, and Hugging Face. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *ACL*, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328. JMLR. org, 2017.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019a.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019b.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL (1)*, 2019.

A Appendix: Introduction to Linguistic Tasks

Task	Type	Example	BERT Small	BERT Base
SVA				
Object Relative Clause	SS	the author that the guard likes [MASK(is/are)] young	1	1
	SP		0.92	0.96
	PS		0.9	0.98
	PP		1	1
Subject Relative Clause	SS	the author that likes the guard [MASK(is/are)] young	1	1
	SP		1	0.96
	PS		1	0.98
	PP		1	1
Within Sentence Complement	SS	the mechanic said the author [MASK(is/are)] young	1	1
	SP		1	1
	PS		1	1
	PP		1	1
Across Prepositional Phrase	SS	the author next to the guard [MASK(is/are)] young	1	0.99
	SP		1	0.98
	PS		0.98	0.98
	PP		1	1
Reflexive Anaphora				
Number Agreement	SS	the author that the guard likes hurt [MASK(himself/themselves)]	0.66	0.6
	SP		0.66	0.74
	PS		0.83	0.83
	PP		1	0.96
Gender Agreement	MM	some wizard who can dress our man can clean [MASK(himself/herself)]	0.78	1
	MF		0.32	0.96
	FF		1	0.9
	FM		0.8	0.66

Table 3: Example of each agreement task and their performance on two BERT models, first 5 tasks are sampled from Marvin & Linzen (2018), the last task is sampled from dataset in Lin et al. (2019), all datasets are constructed as an MLM task according to Goldberg (2019)

B Appendix: Experiment Details

B.1 Convergence of Completeness

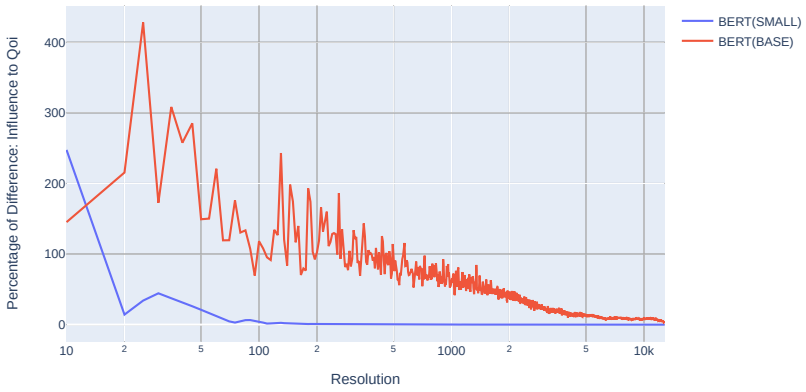
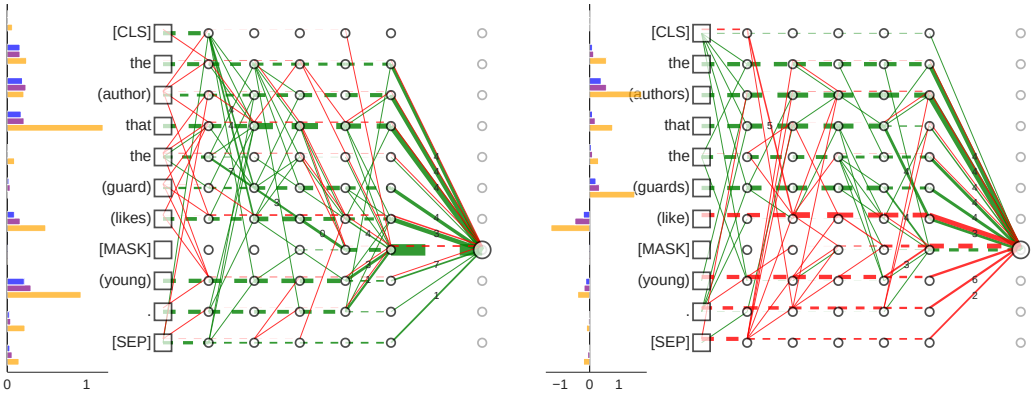


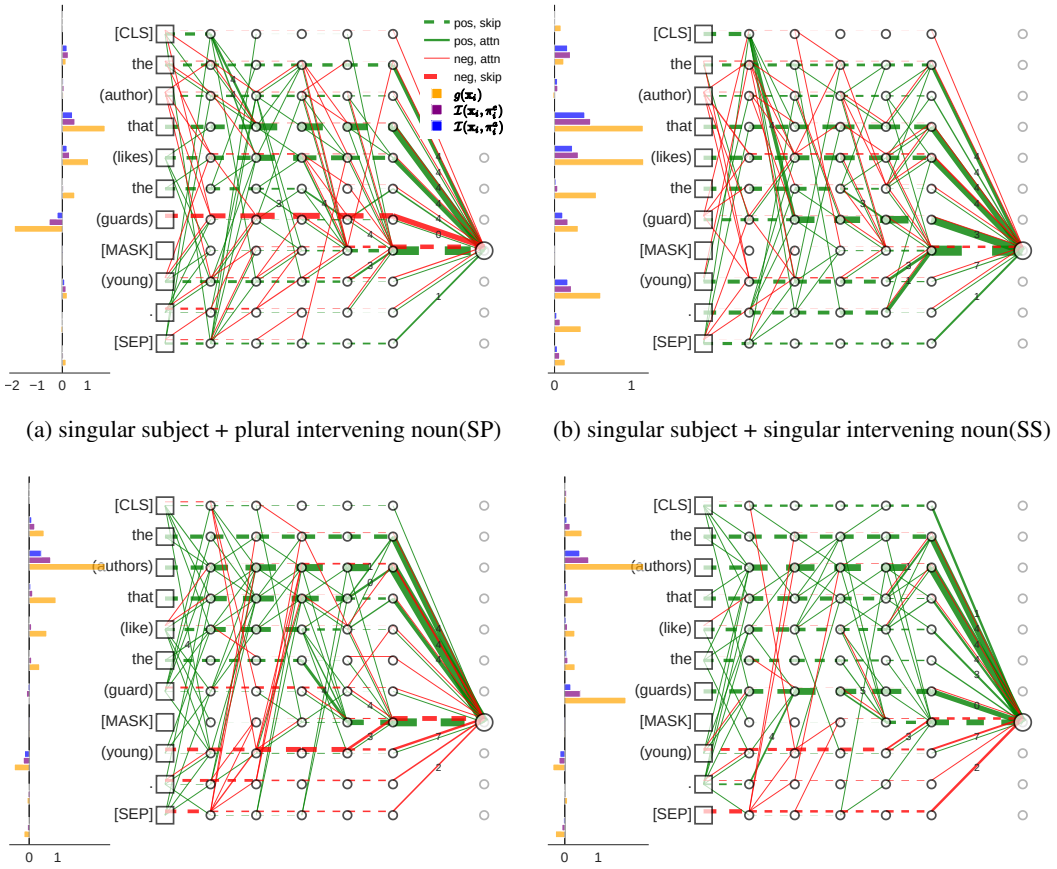
Figure 4: Convergence Analysis for Calculating the Distributional influence(IG) for SVA Across Object Relative Clauses for $BERT(SMALL)$(used in this paper) and $BERT(BASE)$, The much slower convergence of the larger BERT model is likely due to the complicated decision boundaries of larger BERT, masking the output sensitive to small perturbations.

B.2 Influence Graphs for Other Tasks



(a) singular subject + singular intervening noun(SS) (b) plural subject + plural intervening noun(PP)

Figure 5: SVA Across Object Relative Clause.



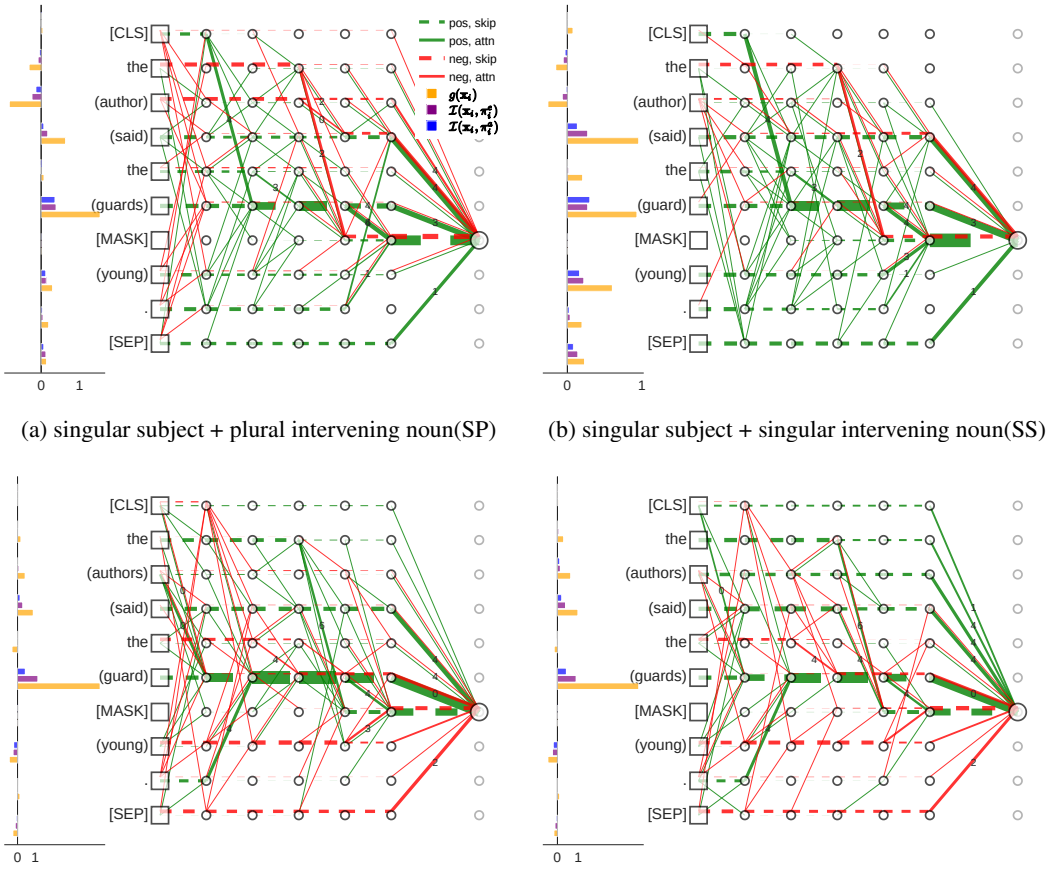
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

Figure 6: SVA Across Subject Relative Clause.



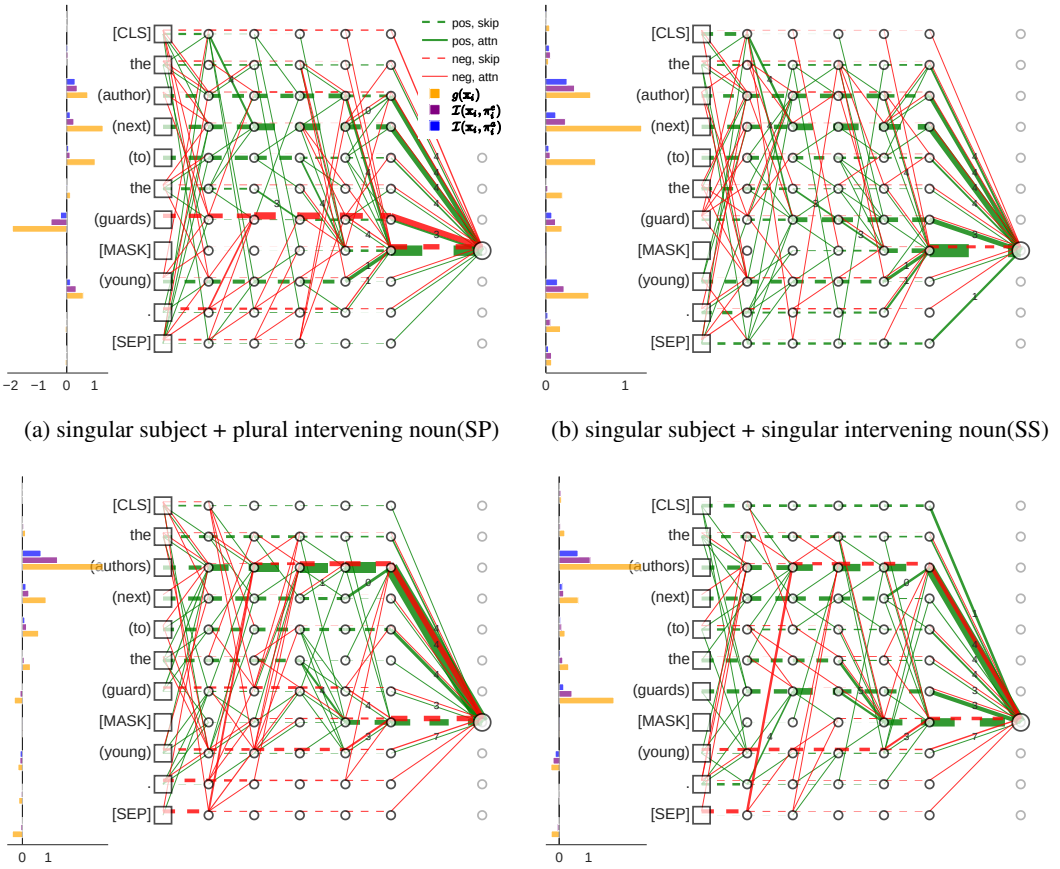
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

Figure 7: SVA Within Sentence Complements.



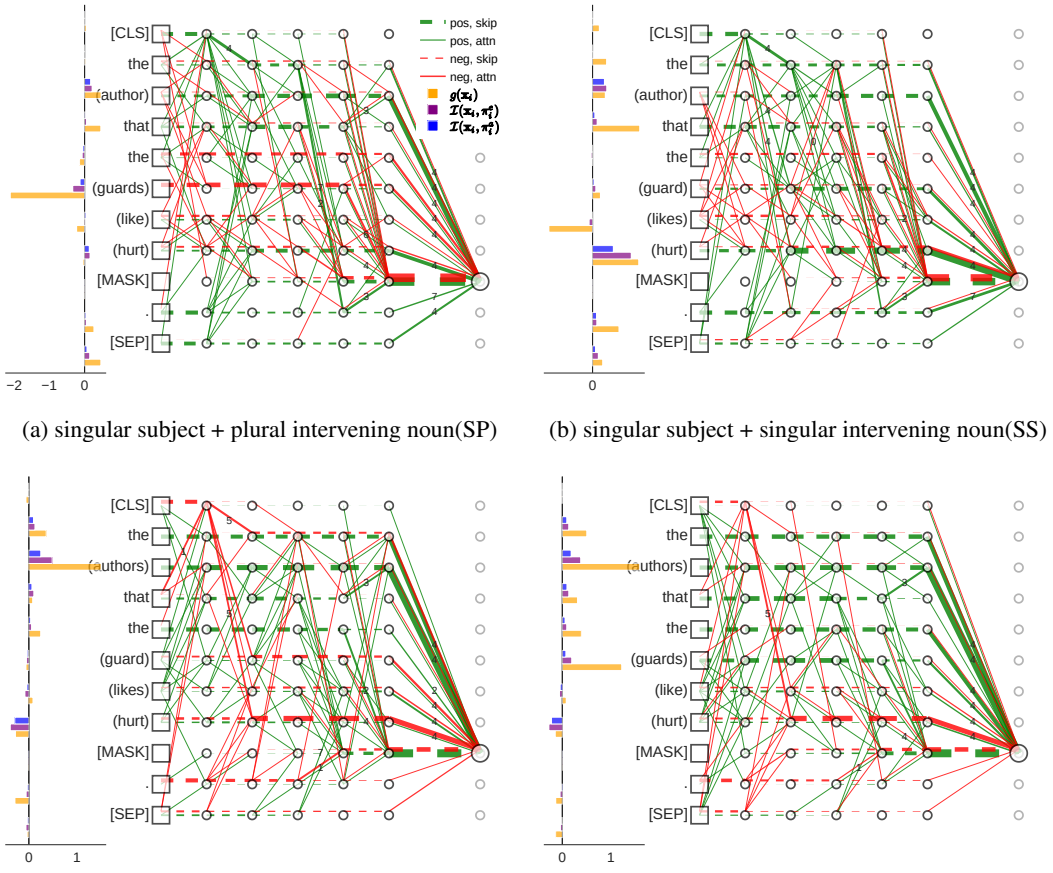
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

Figure 8: SVA Across Prepositional Phrase.



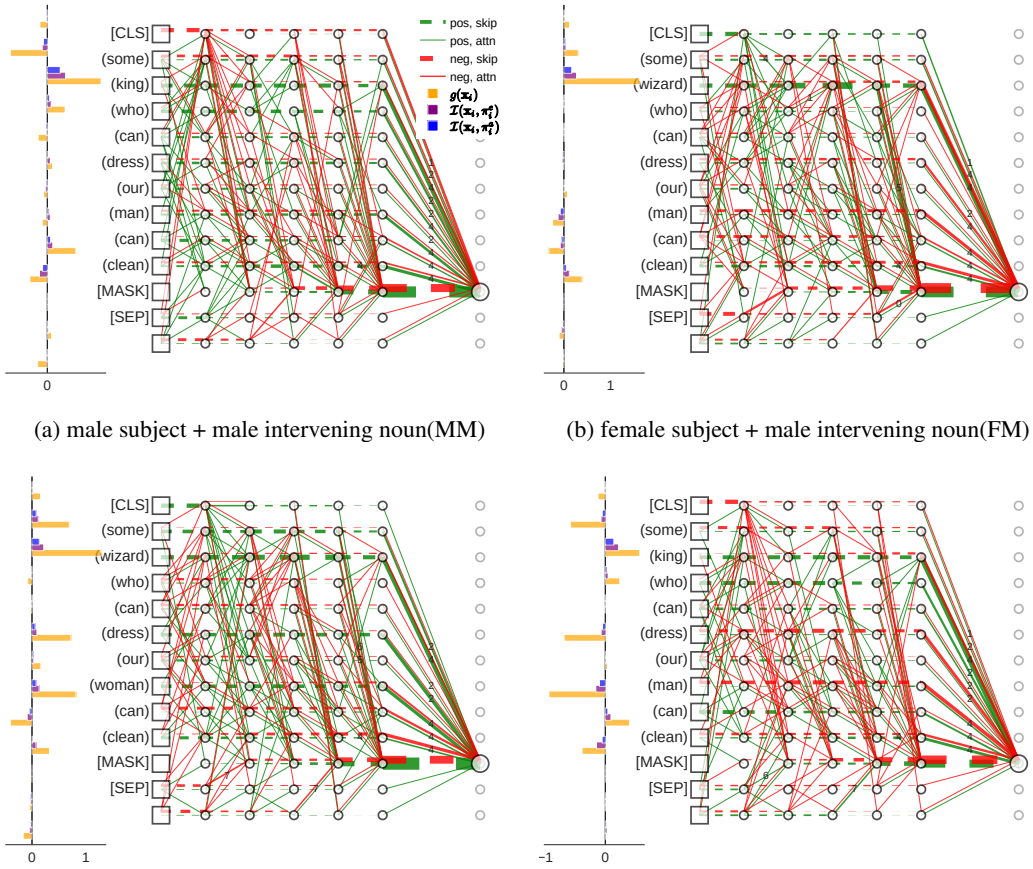
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

Figure 9: RA: Number Agreement



(a) male subject + male intervening noun(MM)

(b) female subject + male intervening noun(FM)

(c) female subject + female intervening noun(FF)

(d) male subject + female intervening noun(MF)

Figure 10: RA: Gender Agreement