Low-rank Subspace for Binding in Large Language Models

Anonymous ACL submission

Abstract

Entity tracking is essential for complex reasoning. To perform in-context entity tracking, language models (LMs) must bind an entity to its attribute (e.g., bind a container to its content) to recall attribute for a given entity. For example, given a context mentioning “The coffee is in Box Z, the stone is in Box M, the map is in Box H”, to infer “Box Z contains the coffee” later, LMs must bind “Box Z” to “coffee”. To explain the binding behaviour of LMs, Feng and Steinhardt (2023) introduce a Binding ID mechanism and state that LMs use a abstract concept called Binding ID (BI) to internally mark entity-attribute pairs. However, they have not directly captured the BI information from entity activations. In this work, we provide a novel view of the Binding ID mechanism by localizing the BI information. Specifically, we discover that there exists a low-rank subspace in the hidden state (or activation) of LMs, that primarily encodes BIs. To identify this subspace, we choose principle component analysis as our first attempt and it is empirically proven to be effective. Moreover, we also discover that when editing representations along directions in the subspace, LMs tend to bind a given entity to other attributes accordingly. For example, by patching activations along the BI encoding direction we can make the LM to infer “Box Z contains the stone” and “Box Z contains the map”.

1 Introduction

The ability of a model to track and maintain information associated with an entity in a context is essential for complex reasoning (Karttunen, 1976; Heim, 1983; Nieuwland and Van Berkum, 2006; Barzilay and Lapata, 2008; Kamp et al., 2010). To recall attribute information in the context, the model must bind entities to their attributes(Feng and Steinhardt, 2023). For example, given Sample 1 and 2, a model must bind the entities (e.g., “Box Z”, “Box M”, “Box H”, “Alex”, “John” and “Carl”) to their attributes (e.g., “coffee”, “stone”, “map”, “bean”, “pie” and “fruit”) respectively so as to accurately recall such as what is in “Box Z” or what is sold by “Alex” without confusion. Binding has also been studied as a fundamental problem in Psychology (Treisman, 1996). To uncover how Language Models (LMs) realize binding in term of internal representation, Feng and Steinhardt (2023) introduce a Binding ID mechanism and state that LMs apply a abstract concept called Binding ID (BI) to bind and mark Entity-Attribute (EA) pairs (e.g., “Box Z” and “coffee”, as shown in Sample 1, where BI is denoted as a numbered square). They also claim that the BI is represented as a vector to be added on the representation (or activation) of a EA pair so that the common vector is used as a key clue to search attribute for a given entity. However, they have not directly captured BI from the activations.

(1) **Context**: The coffee$_{\boxed{0}}$ is in Box Z$_{\boxed{0}}$, the stone$_{\boxed{1}}$ is in Box M$_{\boxed{1}}$, the map$_{\boxed{2}}$ is in Box H$_{\boxed{2}}$.
Query: Box Z$_{\boxed{0}}$ contains the

(2) **Context**: The bean$_{\boxed{0}}$ is sold by Person Alex$_{\boxed{0}}$, the pie$_{\boxed{1}}$ is sold by Person John$_{\boxed{1}}$, the fruit$_{\boxed{2}}$ is sold by Person Carl$_{\boxed{2}}$.
Query: Person Alex$_{\boxed{0}}$ sells the

Since binding is the foundational skill that underlies entity tracking (Feng and Steinhardt, 2023), in this work, we take the entity tracking task (Kim and Schuster, 2023; Prakash et al., 2024) as a benchmark to evaluate the LM’s binding behaviour. Based on the analysis of internal representation on this task, we localize BI information from the activiations and provide a novel view of the Binding ID mechanism. Specifically, we discover that LMs encode (or store) BI information into a low-rank subspace (called BI subspace hereafter), where BI is encoded according to the order of appearance

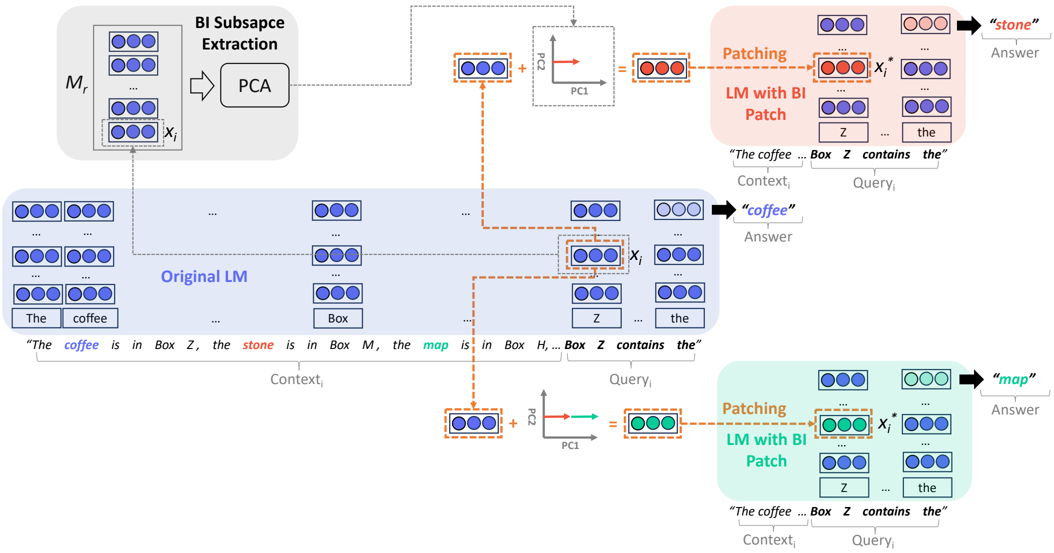


Figure 1: Our main finding on Binding ID (BI) subspace intervention. Patching entity (e.g., "Z") representations along BI direction in activation space yields corresponding changes in model output.

(i.e.,from left to right). To identify the BI subspace, we take Principle Component Analysis (PCA) as our first attempt [1] to capture the subspace representing the BI information, and it is empirically proven be effective. Therefore, our findings confirm and extend prior understanding of BI in LMs.

Going beyond the prior work on BI mechanism (Feng and Steinhardt, 2023), we show that by causally intervening along the BI encoding Principle Component (PC), LMs swap the binding and infer a new attribute for a given entity accordingly. That is, we find a consistent causal relationship between the BI subspace intervention and the inferred attributes by LMs. For example, as shown in Figue 1, by patching activations along a direction (i.e., PC1), we can make the LMs to infer "Box Z contains the stone" and "Box Z contains the map" instead of "Box Z contains the coffee".

Overall, our findings suggest that LMs encodes Binding IDs into a subspace of entity activations in a way that the direction reflects the appearance order (or reversed order) of a EA pair in a given context. Moreover, the discovered BI subspace plays a crucial role in the in-context binding computation. In addition, we find that BI subspace not only exists in the Pretrained large LMs such as Llama2 (Touvron et al., 2023) and Llama3 (AI@Meta, 2024), but also in the code fine-tuned LM such Float-7B (Prakash et al., 2024).

[1]Besids PCA, we also attempt partial least squares regression for capturing BI subspace. Since they achieve similar regression score, we adopt PCA for simplicity. See Appendix (§A.1) for details.

2 Finding Low-rank Subspace for Binding ID

In this section we describe our Principle Component Analysis (PCA) based method to locate the BI subspace in activations of LMs. Firstly we extract entity activation from LMs as shown in Figure 1. Given a LM (e.g., Llama2), and a collection of texts which describe a set of EA pairs for a relation such as Sentence 1 for relation “is_in”, we extract the activation of entity token (e.g., “Z”) in query (denoted as $\mathbf{x}_i$) from certain layer [2] and construct a activation matrix $M_r \in R^{n\times d}$ for a relation r, where n denotes the number of entities and d denotes the dimension of the activation. The row i of M_r is the activation of an entity token (i.e., $\mathbf{x}_i$).

PCA has been applied for identifying various subspace (or direction) such as the subspace encoding language bias (Yang et al., 2021), truth value of assertions (Marks and Tegmark, 2023) and sentiment (Tigges et al., 2023). Inspired by these studies, we choose PCA as our first attempt to localize BI subspace. We hypothesize that in a activation subspace, entities with the same BI tend to cluster together (w.r.t the ones with different BIs), even though these entities have different semantic meaning, and the BIs are encoded as directions (or a PC) in the subspace. For convenience, we number BIs in left-to-right order, and the leftmost BI$=0$.

To capture the subspace, or BI direction, we leverage PCA, which identifies the principle direc-

[2]See Appendix (§A.2) for the layer selection.

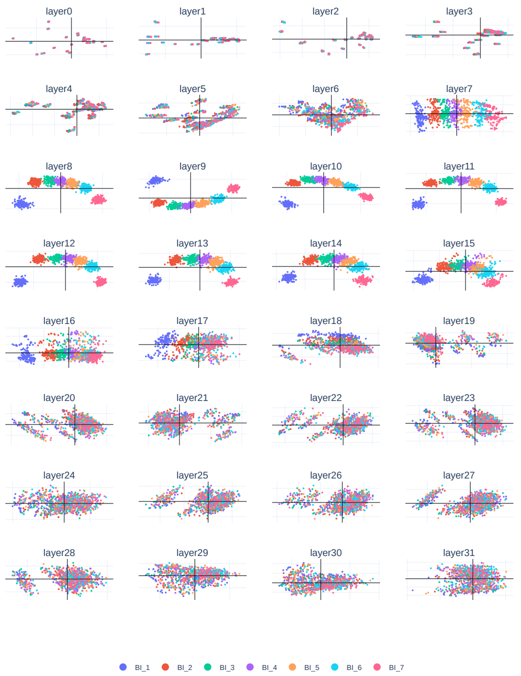


Figure 2: Layer-wise BI subspace visualization on Llama2-7B.

tions of a space. Specifically, the PCA of a activation matrix is $M_r = U_r \Sigma_r V_r^T$, where the columns of $V_r \in R^{d \times d}$ are principle directions of M_r. We takes first c columns of V_r as the BI direction, denoted as $B_r \in R^{d \times c}$.

3 BI Subspace Visualization

We adopt a subset of the entity tracking dataset (Kim and Schuster, 2023; Prakash et al., 2024), which contains $n = 1000$ samples, to create layer-wise activation matrix M_r^l. We then uses the M_r^l to extract the layer-wise BI subspace projection matrix $B_r^l \in R^{d \times 2}$ to visualize the activation. Figure 2 shows the embedding visualization on Llama2-7B, where each point represents the activation of an entity projected via the B_r^l, and the colors represent BIs. From which, we can observe that middle layers, such as layer 8, have a clearly visible direction along which BI increases (or decreases), while the others have tangled distribution. We also observe similar property of distribution on Llama3-8B and Float-7B (§A.3). This indicates that LMs use the middle layers to encode BI information. We call this dimension that represents the order of BI as BI Principle Component (BI-PC). In the following section, we apply causal intervention on the BI-PC to analyze how BI-PC affect the model output.

4 Causal Interventions on Binding ID Subspace

By projecting the activation matrix M_r into the BI subspace, we have found a correlative evidence for the existence of the direction (i.e., BI-PC) that encodes Binding IDs. However, it is possible that the BI information is encoded in the BI subspace but has no effect on model output.

In order to test if Binding IDs are not only encoded in the BI subspace, but that these representations can be steered so as to swap the binding and change LM's output. We now perform interventions to establish the causality. That is, we want to find out if making BI swapping interventions leads to a change in model output.

4.1 Activation Patching

Since LM computation graph could be viewed as causal graph (Meng et al., 2022; McGrath et al., 2023), we intervene on model activations via activation patching (Vig et al., 2020; Wang et al., 2022; Zhang and Nanda, 2023; Heinzerling and Inui, 2024; Engels et al., 2024), and observe the effect on model output. Unlike the common activation patching setup in which one replaces activations resulting from an original input with activations from a corrupted one, we create patches by editing activations along a particular direction (i.e., along BI-PC), similar to the activation editing method of (Matsumoto et al., 2023; Heinzerling and Inui, 2024; Engels et al., 2024). Although automatic methods for localizing model circuits of interest have been proposed (Conmy et al., 2023; Kramár et al., 2024), for simplicity we perform a coarse layer-wise search based on the effect of activation patching in a development set, as shown in Appendix (§A.2), and use the found setting for all experiments.

4.2 Setting

Dataset To explore the internal representation that enables binding, we adopt the entity tracking dataset (Kim and Schuster, 2023; Prakash et al., 2024). The dataset contains English sentence describing a set of objects located in a set of boxes with difference labels, and the task is to infer what is inside a given box. For instance, when a LM is presented with "The coffee is in Box Z, the stone is in Box M, the map is in Box H, ... Box Z contains the", it should infer the next token as "coffee".

Context	Query	Answer for # Step 1	2	3	4	5	6
The coffee is in Box Z, the stone is in Box M, the map is in Box H, the coat is in Box L, the string is in Box T, the watch is in Box E, the meat is in Box F.	Box Z contains the	stone	map	map	string	watch	meat
The letter is in Box Q, the boot is in Box C, the fan is in Box N, the crown is in Box R, the guitar is in Box E, the bag is in Box D, the watch is in Box K.	Box Q contains the	boot	fan	crown	guitar	watch	watch
The cross is in Box Z, the ice is in Box D, the ring is in Box F, the plane is in Box Q, the clock is in Box X, the paper is in Box I, the engine is in Box K.	Box Z contains the	ice	ring	ring	clock	paper	engine

Table 1: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of "r: is_in", where color denotes the BI.

	Template
1	The a_0 is sold by person e_0, ..., the a_i is ..., a_7 is sold by person e_7. Person e_i is selling the
2	The a_0 is applied by person e_0, ..., the a_i is ..., a_7 is applied by person e_7. Person e_i applies the
3	The a_0 is moved by person e_0, ..., the a_i is ..., a_7 is moved by person e_7. Person e_i moved the
4	The a_0 is brought by person e_0, ..., the a_i is ..., a_7 is moved by person e_7. Person e_i brings the
5	The a_0 is pushed by person e_0, ..., the a_i is ..., a_7 is pushed by person e_7. Person e_i pushes the

Table 2: Templates of Dataset.

Each sample involves 7 AE pairs. To evaluate the binding in various context, we also apply the templates shown in Table 2 to generate other 5 datasets with different relation, where a_i and e_i denotes the attribute and entity, and they are sampled from a fixed pool of 224 one-token objects (e.g., "dog", "corn" and "cookie") and 523 of one-token names (e.g., "Alex", "Juli" and "Dan") respectively. We sample $n = 1000$ context from each dataset to run the following analysis.

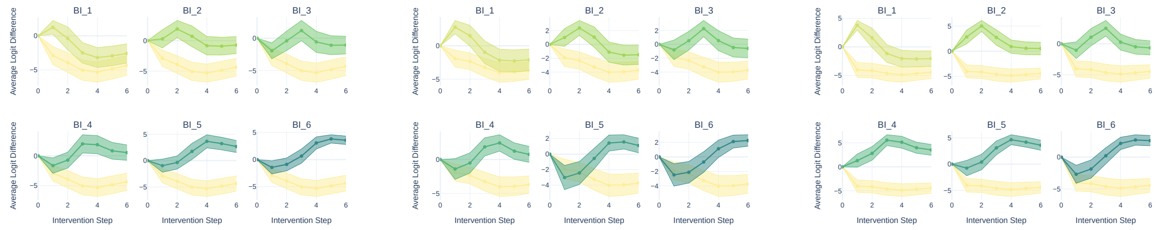
Metrics We apply two evaluation metrics. The logit difference metric introduced in Wang et al. (2022), which calculates difference in logits between original and intervened answers, as well as the "logit flip" accuracy metric (Geiger et al., 2022), which represents the proportion of cases where we alter the model output after a causal intervention.

4.3 Results: Direct Editing BI Subspace

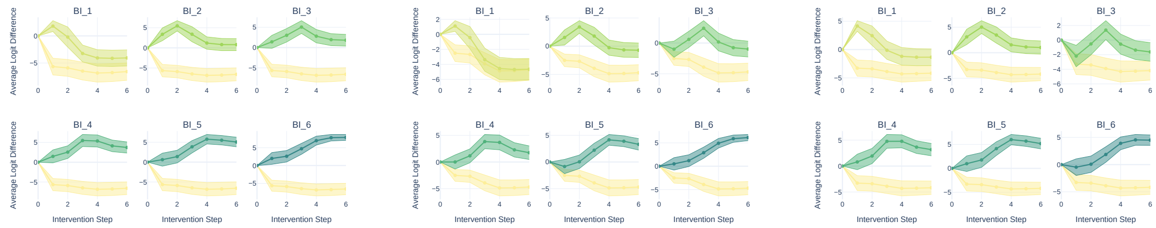
We hypothesize that LMs encode BI information into a low-rank BI subspace. Therefore, we wonder if a LM changes the binding behavior, when adding a particular value v (called step hereafter) along the BI-PC mentioned in Section (§2). For example, if we add one unit of v on the activation of e_0, the LM will reset its BI as 1 and bind attribute a_1 to entity so that infers the a_1 as the attribute of e_0 instead of the original a_0. Similarly, adding two units of v will make the LM infer a_2 for e_0, and so on. We intervene via the Equation 1, where $\mathbf{x}_{0,l}$ is the original activation of e_0 (i.e., the leftmost entity) in layer l, $\mathbf{x}^*_{0,l}$ is the intervened activation, B_r is the BI subspace projection matrix mentioned in Section (§2), α is a hyper-parameter to scale the effect of intervention and β ($0 \leqslant \beta \leqslant 6$) denotes the number of steps.

$$\mathbf{x}^*_{0,l} = \mathbf{x}_{0,l} + \alpha B_r^T (B_r \mathbf{x}_{0,l} + \beta v) \quad (1)$$

Table 1 lists several examples under the BI subspace intervention on the entity tracking dataset (Kim and Schuster, 2023; Prakash et al., 2024). We also list the examples from other datasets in Appendix (§A.5). We can see that when adding 1 step along BI-PC, the model selects "stone" for entity "Z" instead of its original attribute "coffee". Similarly, when the step is doubled, the model will select attribute "map" for the entity, and so on. Although the attribute selection does not strictly follow the number of steps, this indicates, to some extent, that changing the value along BI-PC can induce the swap of attribute.



(a) r: is_in (b) r: sell (c) r: apply



(d) r: move (e) r:bring (f) r: push

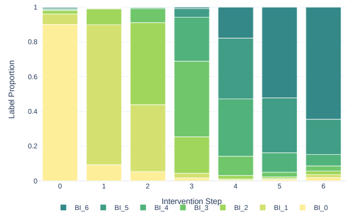
Figure 3: Logit Difference (LD) for BI-PC based intervention across datasets on Llama2-7B, where x axis denotes the number of intervention steps on e_0, y axis does the LD, BI_i represents each target attribute and the light yellow bottom line indicates the LD of original attribute (i.e., a_0). Here, $l = 8$, $v = 2.5$, and $\alpha = 3.0$.

Besides the qualitative analysis, we also conduct quantitative analysis for the causality between the BI subspace based activation patching and the binding behaviour of the LM. We plot mean-aggregated effect of directed activation patching across multiple datasets in Figure 3. Figure 3 indicates how the Logit Difference (LD) of each attribute changes as the step increases. We can observe that as the number of steps increases, LD of the original attribute decreases. In contrast, LD of other attributes gradually increase until a certain point and then gradually decrease. Given a candidate attribute, its peak point generally corresponds the number of steps that equals to its BI. For instance, when adding 3 steps, the points of BI_3 (i.e., attributes of BI$=$ 3) on step$=$ 3 achieve the highest LD score. This indicates that by adjusting the value along the BI-PC, we can increase the probability of the corresponding attribute, thereby swap the answer.

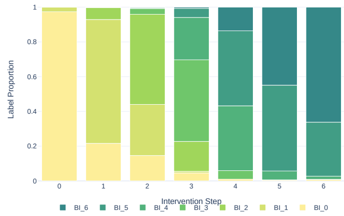
Similarly, Figure 4 illustrates the relation between the number of steps and the logit flip, which gauges the percentage of the predicted attributes under an intervention. Figure 4 shows that as the step increases, the proportion bar becomes darker, it means that the model promotes the proportion of the corresponding following attribute in its inference. For instance, when adding 3 step on the subspace, the a_3 (i.e., BI_3) becomes the major of the answers. This proves that the discovered subspace stores BI information, and the subspace will causally affect the computation of Binding in a LM. See Appendix (§A.8) for the results on Llama3-8B.

4.4 Results: Activation Steering on BI Subspace

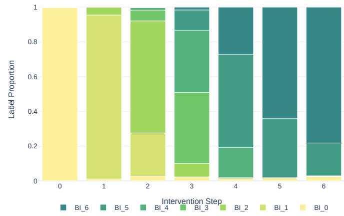
Inspired by the research on Activation Steering (AS) (Turner et al., 2023), we apply an AS method to verify the importance of the BI subspace on LM's binding behaviour. Specifically, we use the following Equation 2 to extract a subspace steering vector $\mathbf{s}_{0\to bi}$, which is proposed to swap BI from 0 to bi, where n is the number of target entities, $\mathbf{x}^i_{bi,l}$ represents the activation of entity e_i from layer l, and its BI is bi. We intervene via Equation 3 and assume that by adding $\mathbf{s}_{0\to bi}$ to the original activation $\mathbf{x}_{0,l}$, we can increase the LD and the proportion of the attribute a_{bi}. Figure 5 shows the results on the entity tracking dataset (Kim and Schuster, 2023; Prakash et al., 2024). (Appendix (§A.6) shows the results on other datasets) These results indicate that AS can achieve the similar tendency as the direct value intervention mentioned in Section (§4.3). For instance, adding $\mathbf{s}_{0\to 3}$, which is used to swap BI form 0 to 3, can increase the LD of a_3 and its proportion in the predicted answers. The consistent tendency with the results of the direct subspace



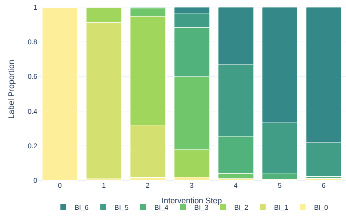
(a) r: is_in



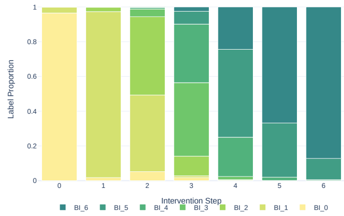
(b) r: sell



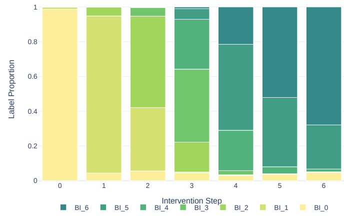
(c) r: apply



(d) r: move



(e) r: bring



(f) r: push

Figure 4: Logit flip for BI-PC based intervention across datasets on Llama2-7B, where x axis denotes the number of intervention steps on e_0, y axis does the proportion of each inferred attribute in model output.

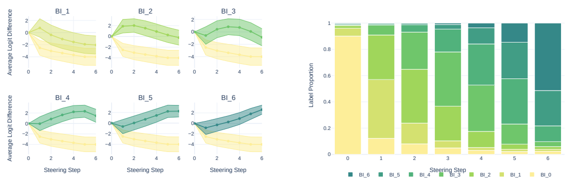


Figure 5: Logit Difference and Logit Flip for activation steering on the entity tracking dataset (i.e., r: is_in), where x axis represents the intervention of $\mathbf{s}_{0\to bi}$.

	Input (original)	
C_1	$a_0^{p0}\, r\, e_0^{p1},\ a_1^{p2}\, r\, e_1^{p3},\ a_2^{p4}\, r\, e_2^{p5}.$	$e_1^{p3}\ r^{-1}$?
C_2	$a_0^{p0}\, r\, e_0^{p1},\ a_1^{p2}\, r\, e_1^{p3},\ a_2^{p4}\, r\, e_2^{p5}.$	$e_2^{p5}\ r^{-1}$?
	Input (with pseudo)	
C_1'	$a_{*0}^{p0}\, r\, e_{*0}^{p1},\ a_{*0}^{p2}\, r\, e_{*0}^{p3},\ a_1^{p4}\, r\, e_1^{p5}.$	$e_1^{p5}\ r^{-1}$?
C_2'	$a_{*0}^{p0}\, r\, e_{*0}^{p1},\ a_1^{p2}\, r\, e_1^{p3},\ a_2^{p4}\, r\, e_2^{p5}.$	$e_2^{p5}\ r^{-1}$?

Table 3: Simplified expression of original inputs and the one modified with pseudo relation, which is proposed to equalize PI for PCA analysis, where $a_0^{p1}\ r\ e_0^{p2}$ represents a relation such as "the apple is in Box C", and e_0^{p2} denotes an entity with BI of 0 and PI of $p2$, $e_2^{p5}\ r^{-1}$? denotes the query on entity e_1, such as "Box C contains the".

editing, shown Figure 3 and Figure 4, further illustrates that the discovered subspace contains the BI information, and more importantly, it plays an important role when the model performs in-context binding computation.

$$\mathbf{s}_{0\to bi} = \frac{1}{n}\sum_{i=1}^{n}(B_r\mathbf{x}_{bi,l}^{i} - B_r\mathbf{x}_{0,l}^{i}) \tag{2}$$

$$\mathbf{x}_{0,l}^{*} = \mathbf{x}_{0,l} + \alpha B_r^{T}\mathbf{s}_{0\to bi} \tag{3}$$

4.5 Binding Subspace and Position

In this section, we discuss the relationship between the BI subspace and Positional Information (PI). As mentioned in Section (§4.3), the discovered subspace stores the BI information, therefore direct intervention on it can swap the answer of LM. However, one counter hypothesis is that the subspace is not used for storing BI information but the PI of attributes, thus the direct intervention merely change the PI so that swap the answer. Regarding the relationship between BI and PI, Feng and Steinhardt (2023) found that even when PI of attributes is swapped, the model still makes correct predictions, thus confirming the independence between BI and PI. Based on this finding, we go one step further and illustrate the independence between the BI subspace and PI. To prove the independence, we create two datasets, one is by extending the original dataset with pseudo relation, as shown in

Input (original)
$a_0^{p0}\ r\ e_0^{p1},\ a_1^{p2}\ r\ e_1^{p3},\ a_2^{p4}\ r\ e_2^{p5}.\quad e_0^{p1}\ r^{-1}$?
Input (with interjection)
$a_0^{p0}\ r\ e_0^{p1},\ j^{p2}\ j\ j^{p3}\ ...j^{pi}\ a_1^{pi+1}\ r\ e_1^{pi+1}...\quad e_0^{p1}\ r^{-1}$?

Table 4: Simplified expression of original input and the one modified with a sequence of interjections, where j^{p3} denotes an interjection j, such as "ah", with position of $p3$, which is also the position of e_1^{p3} in the original input.

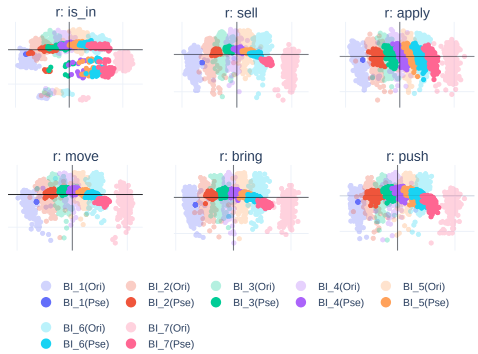


Figure 6: Embedding visualization for activation with equalized PI, where "Ori" denotes the distribution of original dataset, while "Pse" denotes the distribution of the new dataset with pseudo relation.

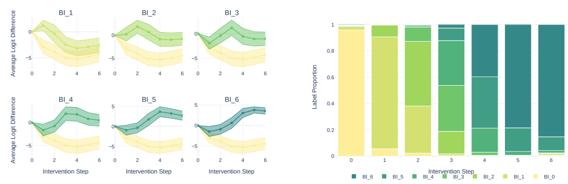


Figure 7: Logit Difference and Logit Flip for activation patching on the dataset of "r: is_in" with interjections. Appendix (§A.9) shows the results on other dataset with the same interjection based modification.

Table 3, and the other is by adding a sequence of interjection, as listed in Table 4.

In Table 3, $a_{*0}^{p0}\ r\ e_{*0}^{p1}$ refers to a pseudo relation, which is a fixed expression, such as "the PC is in Box Z", applied to adjust the PI while keeping the BI. For instance, in Table 3, adding one or two $a_{*0}^{p0}\ r\ e_{*0}^{p1}$ before $a_1\ r\ e_1$ (i.e., C_2' and C_1') does not affect the BI of e_1 but its PI, because e_1 is still the second unique entity from the left (i.e., BI$=1$), but its PI is $p3$ and $p5$ respectively. Using the pseudo relation, we create the data in a manner that the target entity (e.g., e_1^{p5} and e_2^{p5}) to extract activation have the same PI, such as C_1' and C_2' in Table 3.

We apply the method mentioned in Section (§2) on the set of activations $\{\overrightarrow{e_1^{p5}}, \overrightarrow{e_2^{p5}}, ...\}$, where $\overrightarrow{e_1^{p5}}$ denotes the activation of e_1 in $e_1^{p5}\ r^{-1}$?, so as to capture the BI difference and exclude the PI difference, because they share the same PI (i.e., $P5$) but different BI (i.e., 1, 2, ...). Then we compare its BI subspace with the original one (e.g., $\{\overrightarrow{e_1^{p2}}, \overrightarrow{e_2^{p5}}, ...\}$) to analyze how the distribution of BI subspace changes after removing the PI variance. Figure 6 visualizes the BI subspace distribution, where the light colored points denote the original distribution, and the dark ones are from the new one with equalized PI. We can observe that after removing the PI difference, the distribution is still similar to the original one that there is a clearly visible direction along which BI increases. This illustrates that our PCA based method can capture BI information, that is, along the direction of BI-PC, and it does not causally depend on PI. See Appendix (§A.4) for our further analysis on how the context of binding affects the distribution.

Another dataset is created by adding a sequence of interjections after the first attribute entity pair, as illustrated in Table 4. Since there is no BI information in the interjection (e.g., j^{p3}), adding it only changes the PI of its following entities and attributes. We set the number of interjections as that its last PI is larger than the last PI of its original input (e.g., $pi > p5$ in Table 4). Based on this dataset, we conduct the same intervention on its BI subspace, as mentioned in Section (§4.3). The counter argument is that the subspace only captures PI, and the intervening step only changes the PI information. Specifically, adding one unit of v on $\overrightarrow{e_0^{p1}}$ can change its PI from $p1$ to $p3$, which is the PI of e_1, and its attribute is a_1, as shown in Table 4, thereby the LM swaps the answer from a_0 to a_1. If it is true, then the same intervention will not change the answer on the new dataset, because following the counter argument, after adding one unit of v on $\overrightarrow{e_0^{p1}}$, its PI becomes $p3$, and $p3$ is the PI of j^{p3}. The LM thus would not select a_1 as its answer. However, the results on Figure 7 shows that the subspace intervention on the new dataset achieves similar results as the original one, as shown in Figure 3 and Figure 4, proving the counter argument wrong and indicating the independence between BI subspace and PI.

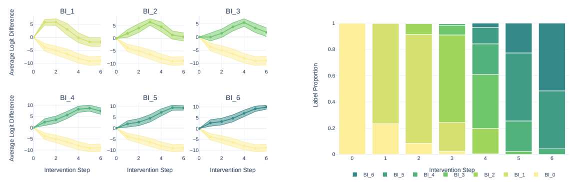


Figure 8: Logit Difference and Logit Flip for activation patching on the entity tracking dataset (i.e., r: is_in) on Float-7B. See Appendix (§A.7) for the results on other datasets.

4.6 Results on Fine-Tuned LM

Kramár et al. (2024); Kim et al. (2024) claim that the code fine-tuned LM, such as Float-7B (Prakash et al., 2024) outperforms the pretrained LM on the entity tracking task (Kim and Schuster, 2023; Prakash et al., 2024). Since the code fine-tuned LM performs well on the entity tracking task that requires the BI subspace based computation, we hypothesize that BI subspace also exists in the code fine-tuned LM and the intervention along BI-PC will causally affect the model output. To prove the hypothesis, we conduct the intervention on Float-7B and show results in Figure 8. We found that the BI subspace based intervention on Float-7B achieves the similar results as on Llama2-7B, indicating that the BI subspace not only exists in the pretrained LM but also in the fine-tuned one. In addition, adding the same step value (i.e., v) on Float-7B will achieve higher LD value than Llama2-7B, indicating that the code fine-tuned LM is more sensitive to the BI subspace based intervention. For instance, the maximum LD of a_4 in the former is around 10, and it is 2 times larger than the one in the latter, which is around 5. This might partially explains why the code fine-tuned LM performs better than the original one, because code fine-tuning might enhance the function of BI subspace so that it is more sensitive on the intervention and more effective on the in-context entity tracking task.

5 Related Work

Linear Representation Recent research found that sequence models trained only on next token prediction linearly represent various semantic concepts including Othello board positions (Li et al., 2022; Nanda et al., 2023), the truth value of assertions (Marks and Tegmark, 2023), sentiment (Tigges et al., 2023), and numeric values such as elevation, population, birth year, and death year (Gurnee and Tegmark, 2023; Heinzerling and Inui, 2024). Continuing this line of research, in this work, we discover that LMs such as Llama-2 can also linearly encode BI, because there is a linear direction that primarily encodes BI in the activations.

Knowledge Localization Many works aim to localize and edit factual relations (e.g., "capital of") that LMs learn from pretraining and are stored into model weights (Geva et al., 2020; Dai et al., 2021; Meng et al., 2022; Geva et al., 2023; Hernandez et al., 2023). Different from this line of research, this work studies in-context representations of relations and analyzes how they are stored in model activations.

Mechanistic Interpretability Notable progress has been made in uncovering circuits performing various tasks within LMs (Elhage et al., 2021; Wang et al., 2022; Wu et al., 2024). Recently, Prakash et al. (2024) identify the circuit for entity tracking task. Feng and Steinhardt (2023) introduce a Binding ID Mechanism for explaining the binding problem, state that LMs use the abstract concept BI to internally mark entity-attribute pairs. However, they does not directly capture BI information from activations. Therefore, they have not answered how LMs store the BI information into entity activations, how to localize the BI information and whether the localized BI information causally affect the model binding behaviour.

6 Conclusion and Future Work

In this work, we study the in-context binding, a fundamental skill underlying many complex reasoning and natural language understanding tasks. We provide a novel view of the Binding ID mechanism introduced by Feng and Steinhardt (2023). We discover that there exists a low-rank subspace in the hidden state (or activation) of LMs, that primarily encodes BIs. What is more, we also discover that when editing representations along BI-PC in the subspace, LMs tend to bind a given entity to other attributes accordingly. Our future work includes: 1. the analysis of BI subspace in the setting of multiple predicates instead of the single one (e.g., "r: is_in"); 2. the study of interaction between in-context binding and factual knowledge learned from pretraining; 3. BI subspace based mechanistic analysis.

Limitation

The limitations of our research include the following points. Firstly, we only analyze BI subspace on the attribute prediction task, but not on the entity inference task (i.e., given an attribute to infer its entity). Secondly, we lack the analysis on how predicate (or relation) affect the BI subspace, and how the results of BI-subspace based intervention differ with the type of predicate. Thirdly, although we use a publicly available entity tracking dataset, it is still a synthesized dataset. Therefore, for uncovering how LMs bind and track entity in reality, it is necessary to analyze the BI subspace on a real world dataset. The last but not lest, we only analyze binding from the perspective of representation and localize BI subspace. However, we have not answered what is the mechanism that generates the subspace and what is the circuit that utilizes the subspace for binding.

Ethical Statement

The existing dataset (Kim and Schuster, 2023; Prakash et al., 2024) and LMs (i.e., Llama2-7B, Float-7B and Llama3-8B) are applied according to their intended research purpose. The synthetic datasets we adopted in this work are automatically created by strictly following the rule (or pattern) of the existing dataset, where the entities and attributes are sampled from a pool of wide variety of one-token names and concepts. Therefore, there is no ethical concern on human annotation bias and semantic biases. The datasets and code will be publicly available to ensure the reproducibility of our experiments.

References

AI@Meta. 2024. Llama 3 model card.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. Computational Linguistics, 34(1):1–34.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. Advances in Neural Information Processing Systems, 36:16318–16352.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. arXiv preprint arXiv:2104.08696.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. Transformer Circuits Thread, 1:1.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. 2024. Not all language model features are linear.

Jiahai Feng and Jacob Steinhardt. 2023. How do language models bind entities in context? arXiv preprint arXiv:2310.17191.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In International Conference on Machine Learning, pages 7324–7338. PMLR.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. arXiv preprint arXiv:2304.14767.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. arXiv preprint arXiv:2012.14913.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. arXiv preprint arXiv:2310.02207.

Irene Heim. 1983. File change semantics and the familiarity theory of definiteness. Semantics Critical Concepts in Linguistics, pages 108–135.

Benjamin Heinzerling and Kentaro Inui. 2024. Monotonic representation of numeric properties in language models. arXiv preprint arXiv:2403.10381.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2023. Linearity of relation decoding in transformer language models. arXiv preprint arXiv:2308.09124.

Hans Kamp, Josef Van Genabith, and Uwe Reyle. 2010. Discourse representation theory. In Handbook of Philosophical Logic: Volume 15, pages 125–394. Springer.

Lauri Karttunen. 1976. Discourse referents. In Notes from the linguistic underground, pages 363–385. Brill.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. arXiv preprint arXiv:2305.02363.

Najoung Kim, Sebastian Schuster, and Shubham Toshniwal. 2024. Code pretraining improves entity tracking abilities of language models. arXiv preprint arXiv:2405.21068.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. arXiv preprint arXiv:2403.00745.

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

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. arXiv preprint arXiv:2310.06824.

Yuta Matsumoto, Benjamin Heinzerling, Masashi Yoshikawa, and Kentaro Inui. 2023. Tracing and manipulating intermediate values in neural math problem solvers. arXiv preprint arXiv:2301.06758.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. arXiv preprint arXiv:2307.15771.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. Advances in Neural Information Processing Systems, 35:17359–17372.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. arXiv preprint arXiv:2309.00941.

Mante S Nieuwland and Jos JA Van Berkum. 2006. When peanuts fall in love: N400 evidence for the power of discourse. Journal of cognitive neuroscience, 18(7):1098–1111.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. arXiv preprint arXiv:2402.14811.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. arXiv preprint arXiv:2310.15154.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Anne Treisman. 1996. The binding problem. Current opinion in neurobiology, 6(2):171–178.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. arXiv preprint arXiv:2308.10248.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. Advances in neural information processing systems, 33:12388–12401.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. arXiv preprint arXiv:2211.00593.

Svante Wold, Michael Sjöström, and Lennart Eriksson. 2001. Pls-regression: a basic tool of chemometrics chemometr. Intell. Lab, 58(2):109–130.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2024. Interpretability at scale: Identifying causal mechanisms in alpaca. Advances in Neural Information Processing Systems, 36.

Ziyi Yang, Yinfei Yang, Daniel Cer, and Eric Darve. 2021. A simple and effective method to eliminate the self language bias in multilingual representations. arXiv preprint arXiv:2109.04727.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. arXiv preprint arXiv:2309.16042.

A Appendix

A.1 Partial least squares regression and PCA

Besides PCA, a commonly used unsupervised Dimension Reduction (DR) method, we also attempt Partial Least Squares regression (PLS) (Wold et al., 2001), a supervised DR method. PLS extracts a set of ordered latent variables that maximizes the co-variability between the features (e.g., activations) and the scores to be predicted (e.g., BI). We perform PCA and PLS on a development set and compare their regression cures in Figure 9. We can observe that both the first PCA component and the first PLS direction contain almost all information about BI of target entity, because their regression score is close to one. The consistency indicates that PCA is an effective method to capture BI.

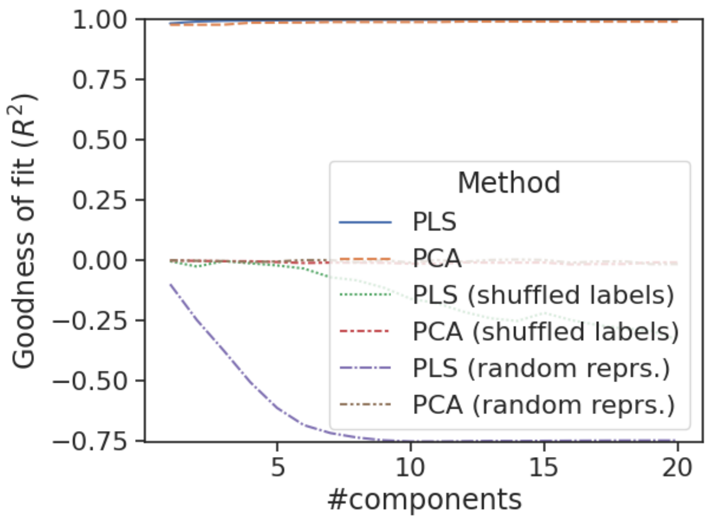
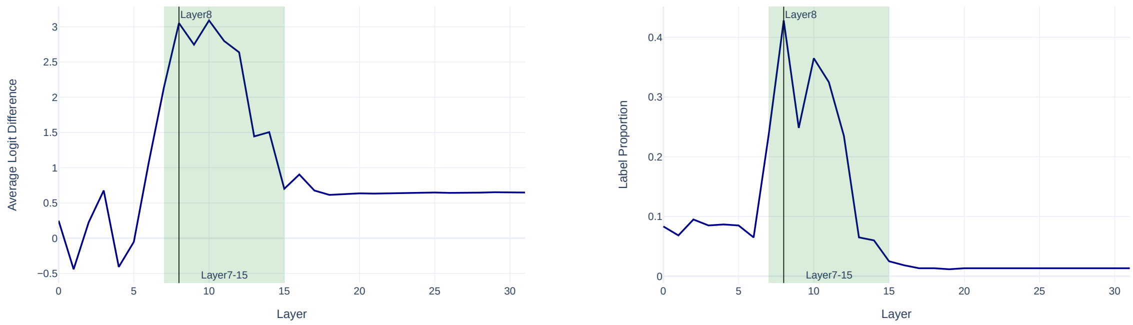


Figure 9: Regression curves for PLS and PCA.

A.2 Layer-wise Intervention

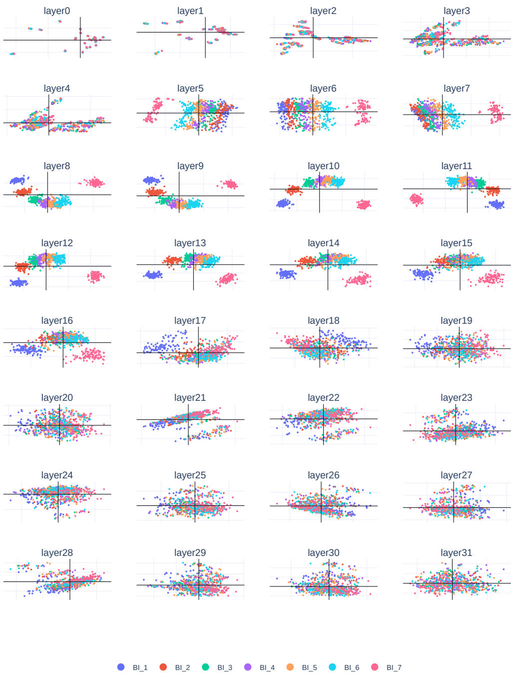
To determine how well BI subspace from different layers of a LM causally affects the model output, we perform layer-wise BI-PC based intervention mentioned in Section (§4.3) on our development set. In Figure 10, we can observe that BI subspace from middle layers (i.e., from layer7 to layer15, especially layer8) significantly affect the computation of binding, and interestingly, these layers also overlap with the ones that clearly encode BI information, as shown in Figure 2. Based on such analysis, we select the layer to perform activation patching.



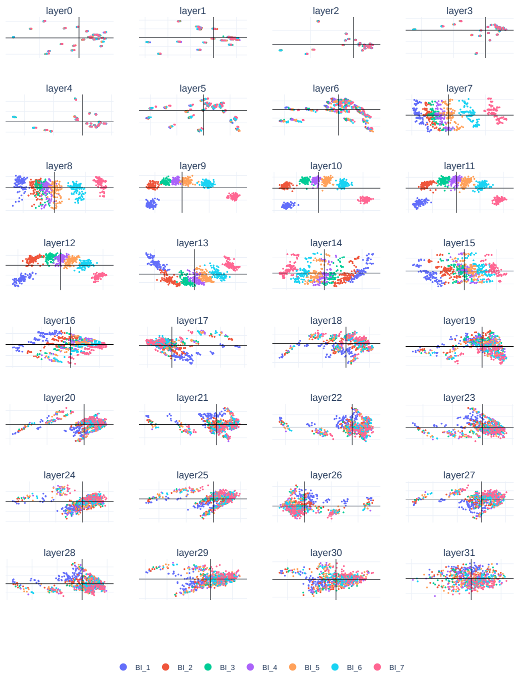
(a) (b)

Figure 10: Average Logit Difference (LD) and logit flip for layer-wise BI-PC based intervention on Llama2-7B, where x axis denotes the layer, the colored zone indicates the layers that are sensitive to the intervention, and the vertical line represents the most sensitive one (i.e., Layer 8), Y axis denotes the average LD and the proportion of inferred attributes (excluding the original one) in Figure 10a and Figure 10b respectively.

A.3 Layer-wise Embedding Visualization on Llama3-8B and Float-7B



(a) Llama3-8B

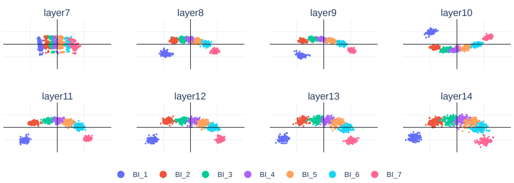


(b) Float-7B

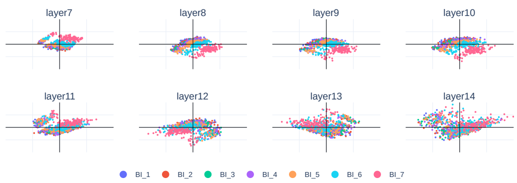
Figure 11: Layer-wise BI subspace visualization on Llama3-8B and Float-7B.

A.4 Layer-wise Embedding Visualization after Masking Binding Information

The counter argument is that the captured subspace only represents the positional information. To test the claim, we attempt to mask the context around the entities and attributes with random two-letter tokens (e.g., “td”) so as to ablate the binding information and keep the positional information. For instance, we convert “the document is in Box Q , the bus is in Box F, . . . ” as “pl document td cy wa Q br fl bus ti eq fs F . . . ” so that there is no relational information in the latter. Figure 12 compares the BI subspace distribution between with and without binding information. We can observe that ablation of binding information tangles the distribution so that there is no clear clustering for each BI (e.g., by comparing layer14). This indicates that our discovered subspace encodes binding information.



(a) w/ Binding Information



(b) w/o Binding Information

Figure 12: Layer-wise BI subspace visualization for w/ and w/o binding information on Llama2-7B.

A.5 Case Study on Llama2-7B

Context	Query	Answer for # Step					
		1	2	3	4	5	6
The bugis sold by person Esta, the spawn is sold by person Fritz, the wine is sold by person Inga, the paste is sold by person Ward, the poison is sold by person Albert, the crow is sold by person Davis, the nest is sold by person Val .	Person Esta is selling the	spawn	wine	paste	poison	crow	nest
The virus is sold by person Anna, the fur is sold by person Earl, the pill is sold by person Flor, the bean is sold by person Roy, the spawn is sold by person Kam, the farm is sold by person Young, the sheep is sold by person Billy.	Person Anna is selling the	fur	pill	spawn	spawn	farm	sheep
The root is sold by person Carl, the mouse is sold by person Marco, the fruit is sold by person Luke, the bug is sold by person Paul, the grass is sold by person Inga, the pie is sold by person Pok, the cookie is sold by person George.	Person Carl is selling the	mouse	fruit	bug	grass	cookie	cookie

Table 5: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of "r: sell", where color denotes the BI.

Context	Query	Answer for # Step					
		1	2	3	4	5	6
The carbon is applied by person Wei, the liquid is applied by person Season, the bath is applied by person Robert, the fog is applied by person Daniel, the heavy is applied by person Roma, the motor is applied by person Ara, the pool is applied by person Jorge	Person Wei applies the	liquid	bath	fog	motor	motor	pool
The rain is applied by person Kurt, the gauge is applied by person Jon, the dust is applied by person Newton, the jet is applied by person Dan, the floor is applied by person Alfred, the low is applied by person Mike, the basket is applied by person April	Person Kurt applies the	gauge	dust	jet	floor	basket	basket
The lamp is applied by person Angel, the bucket is applied by person Carl, the canvas is applied by person Bert, the cargo is applied by person Otto, the plain is applied by person Johnny, the floor is applied by person John, the heavy is applied by person Era.	Person Angel applies the	bucket	canvas	cargo	plain	floor	heavy

Table 6: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of "r: apply", where color denotes the BI.

Context	Query	Answer for # Step					
		1	2	3	4	5	6
The lip is moved by person Mack, the tract is moved by person Sommer, the pen is moved by person Son, the tip is moved by person August, the bat is moved by person Monte, the socket is moved by person Marco, the hook is moved by person Paul.	Person Mack moved the	tract	pen	tip	bat	hook	hook
The mask is moved by person Jules, the timer is moved by person Ward, the bullet is moved by person Ana, the eye is moved by person Val, the button is moved by person Andy, the lock is moved by person Arnold, the colon is moved by person Betty.	Person Jules moved the	timer	bullet	button	lock	lock	colon
The mask is moved by person Cole, the neck is moved by person Donald, the pad is moved by person Beth, the cone is moved by person Jorge, the tail is moved by person Lou, the thread is moved by person Alfred, the toe is moved by person Edward.	Person Cole moved the	neck	pad	cone	tail	toe	toe

Table 7: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of “r: move”, where color denotes the BI.

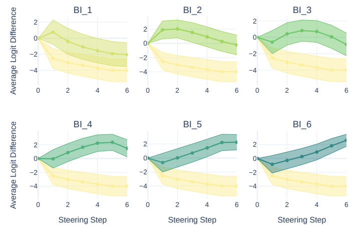
Context	Query	Answer for # Step					
		1	2	3	4	5	6
The creature is brought by person Tam, the guitar is brought by person Frank, the dress is brought by person Stuart, the block is brought by person Victor, the brain is brought by person David, the coffee is brought by person Mack, the radio is brought by person Roger.	Person Tam brings the	guitar	dress	block	brain	coffee	radio
The boat is brought by person Luke, the pipe is brought by person Clara, the pot is brought by person Han, the bill is brought by person Chi, the milk is brought by person Scott, the card is brought by person Henry, the brick is brought by person Morris	Person Luke brings the	pipe	pot	bill	card	brick	brick
The fan is brought by person Van, the note is brought by person Clara, the block is brought by person Alex, the newspaper is brought by person Peg, the crown is brought by person Jan, the car is brought by person Pok, the magnet is brought by person Golden.	Person Van brings the	note	block	crown	car	magnet	magnet

Table 8: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of “r: bring”, where color denotes the BI.

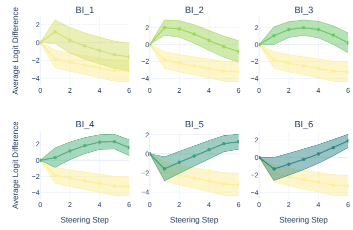
Context	Query	Answer for # Step					
		1	2	3	4	5	6
The load is pushed by person Mike, the atom is pushed by person Mira, the tin is pushed by person Juli, the stud is pushed by person Sam, the sedan is pushed by person Pia, the bath is pushed by person Leo, the growth is pushed by person Pat.	Person Mike pushes the	atom	tin	stud	bath	growth	growth
The mud is pushed by person Thomas, the heavy is pushed by person Ralph, the tile is pushed by person Pierre, the import is pushed by person Perry, the arm is pushed by person Robert, the lung is pushed by person Kurt, the cabin is pushed by person Ernest.	Person Thomas pushes the	heavy	tile	import	arm	cabin	cabin
The bed is pushed by person Fran, the lever is pushed by person Lan, the cord is pushed by person Paris, the vent is pushed by person Gene, the thumb is pushed by person Marie, the mouth is pushed by person Asia, the ear is pushed by person Lang.	Person Fran pushes the	lever	cord	vent	thumb	thumb	ear

Table 9: Attributes inferred by Llama2-7B as a result of directed activation patching along BI-PC in the BI subspace on the dataset of “r: push”, where color denotes the BI.

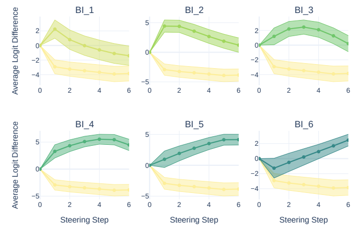
A.6 Activation Steering on Llama2-7B



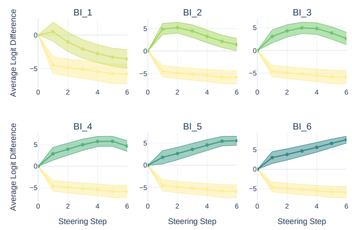
(a) r: is_in



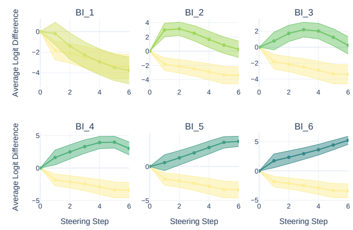
(b) r: sell



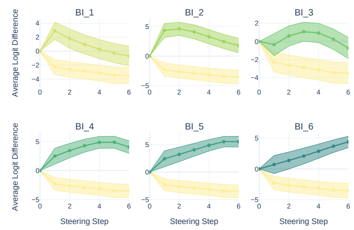
(c) r: apply



(d) r: move

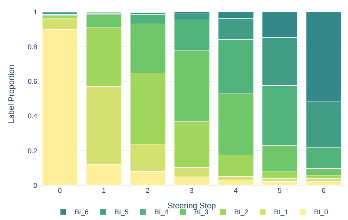


(e) r:bring

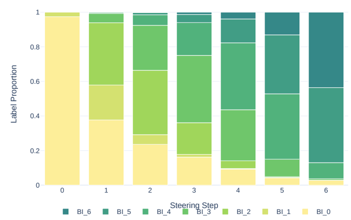


(f) r: push

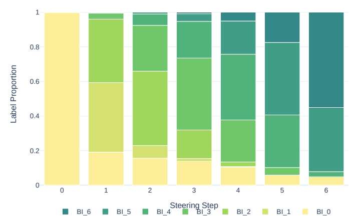
Figure 13: Logit Difference (LD) for BI subspace based activation steering across datasets on Llama2-7B, where x axis represents the intervention of $\mathbf{s}_{0\to bi}$ on the activation of e_0. Here, $l = 8$ and $\alpha = 1.25$.



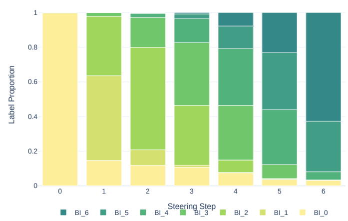
(a) r: is_in



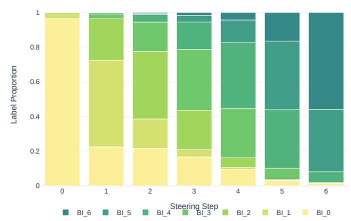
(b) r: sell



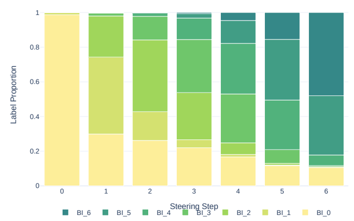
(c) r: apply



(d) r: move



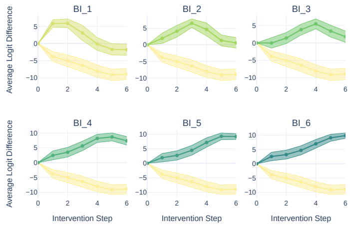
(e) r: bring



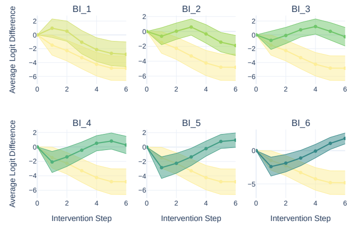
(f) r: push

Figure 14: Logit flip for BI subspace based activation steering across datasets on Llama2-7B, where x axis represents the intervention of $\mathbf{s}_{0\to bi}$ on the activation of e_0.

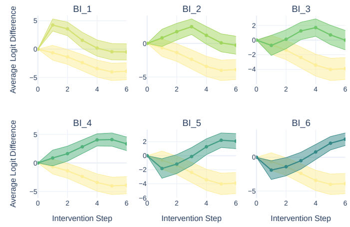
A.7 Activation Patching on Float-7B



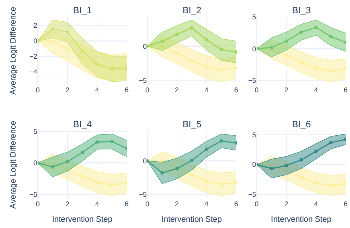
(a) r: is_in



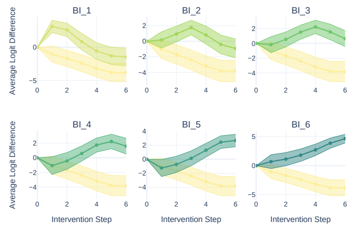
(b) r: sell



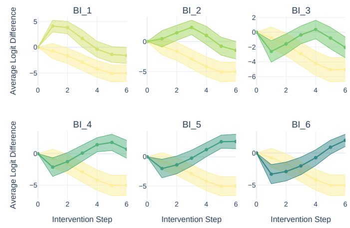
(c) r: apply



(d) r: move

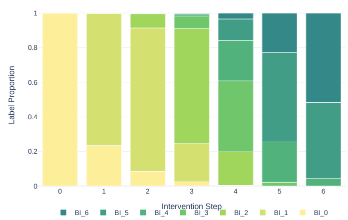


(e) r:bring

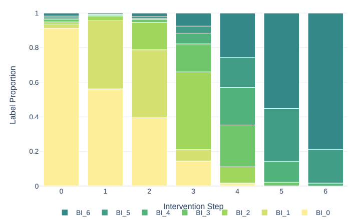


(f) r: push

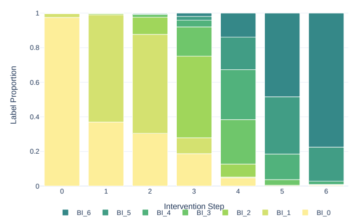
Figure 15: Logit Difference (LD) for BI-PC based intervention across datasets on Float-7B, where x axis denotes the number of intervention steps on e_0, y axis does the LD, BI_i represents each target attribute and the light yellow bottom line indicates the LD of original attribute (i.e., a_0). Here, $l = 10$, $v = 2.55$, and $\alpha = 5.0$.



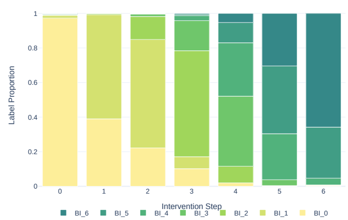
(a) r: is_in



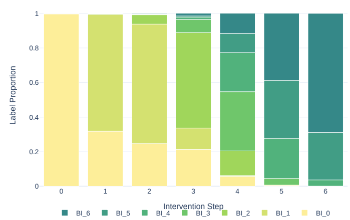
(b) r: sell



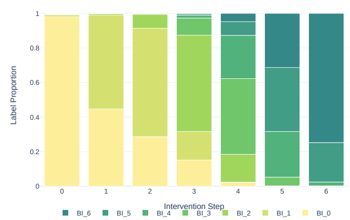
(c) r: apply



(d) r: move



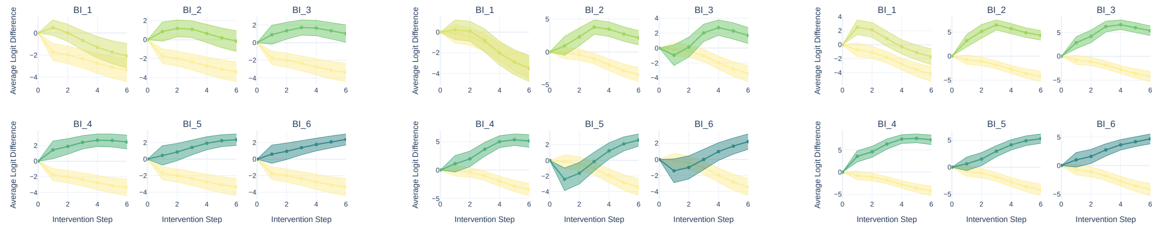
(e) r: bring



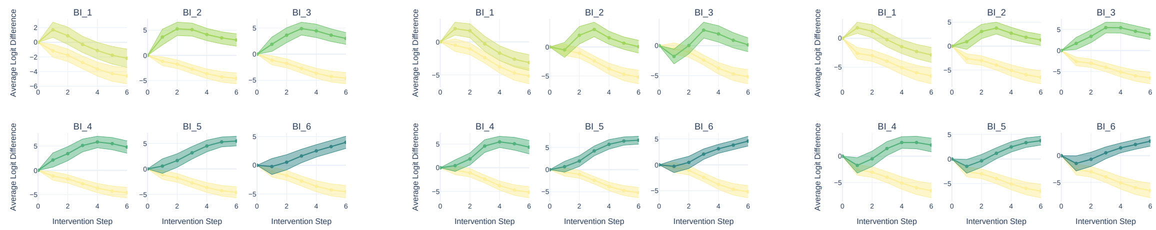
(f) r: push

Figure 16: Logit flip for BI-PC based intervention across datasets on Float-7B, where x axis denotes the number of intervention steps on e_0, y axis does the proportion of each inferred attribute in model output.

A.8 Activation Patching on Llama3-8B

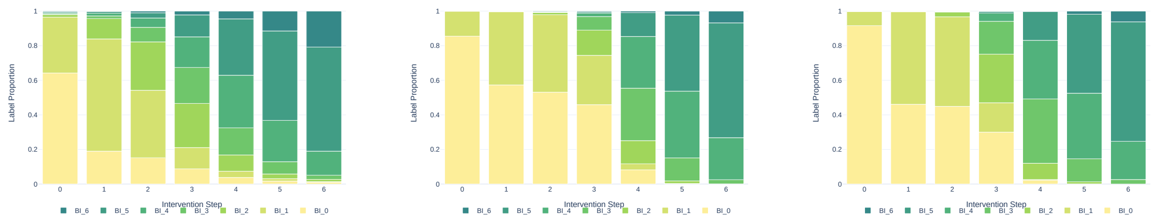


(a) r: is_in (b) r: sell (c) r: apply

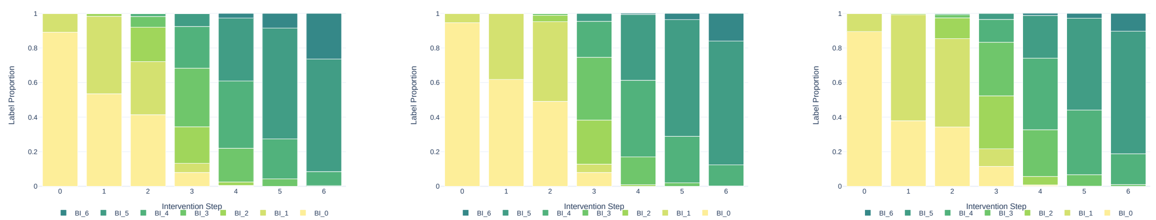


(d) r: move (e) r:bring (f) r: push

Figure 17: Logit Difference (LD) for BI-PC based intervention across datasets on Llama3-8B, where x axis denotes the number of intervention steps on e_0, y axis does the LD, BI_i represents each target attribute and the blue line indicates the LD of original attribute (i.e., a_0). Here, $l = 10$, $v = 0.65$, and $\alpha = 2.0$.



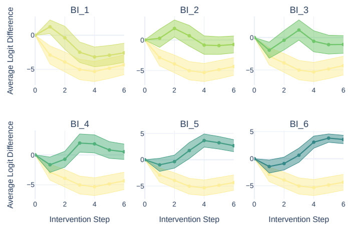
(a) r: is_in (b) r: sell (c) r: apply



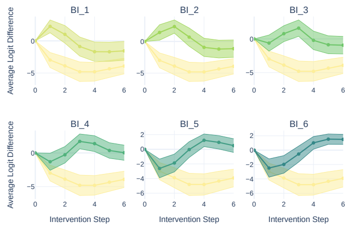
(d) r: move (e) r: bring (f) r: push

Figure 18: Logit flip for BI-PC based intervention across datasets on Llama3-8B, where x axis denotes the number of intervention steps on e_0, y axis does the proportion of each inferred attribute in model output.

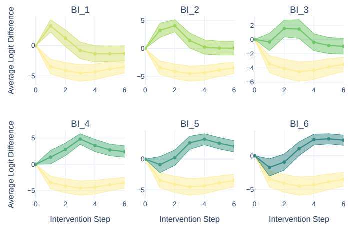
A.9 Activation Patching on the New Dataset with Interjections



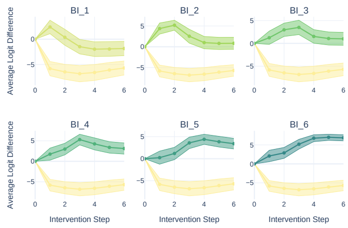
(a) r: is_in



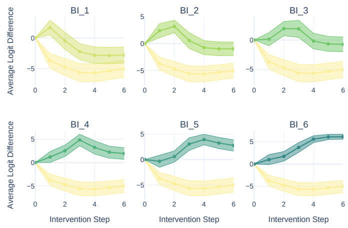
(b) r: sell



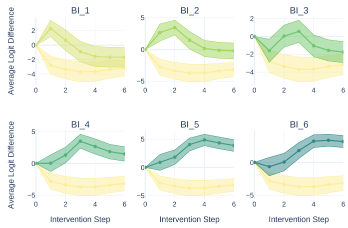
(c) r: apply



(d) r: move

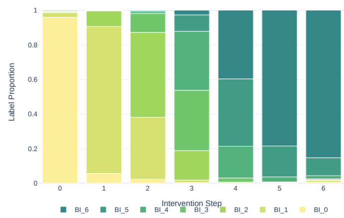


(e) r:bring

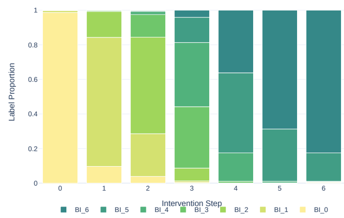


(f) r: push

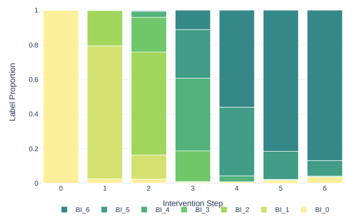
Figure 19: Logit Difference for activation patching on the dataset with interjections. Here, $l = 8$, $v = 2.5$, and $\alpha = 3.0$.



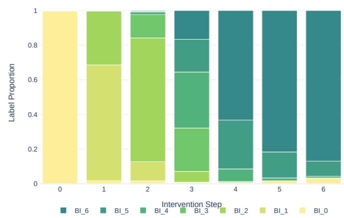
(a) r: is_in



(b) r: sell

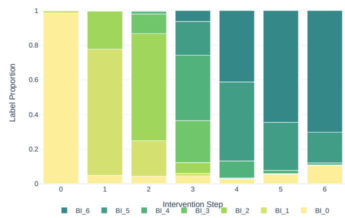


(c) r: apply



(d) r: move

(e) r: bring



(f) r: push

Figure 20: Logit Flip for activation patching on the dataset with interjections.