THE UNIVERSITY OF CHICAGO

GEOMETRIC AND ALGEBRAIC STRUCTURES IN FOUNDATION MODEL REPRESENTATIONS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

YIBO JIANG

CHICAGO, ILLINOIS

AUGUST 2025

Copyright © 2025 by Yibo Jiang

All Rights Reserved

“The important thing isn’t can you read music, it’s can you hear it.

Can you hear the music, Robert?”

— *Oppenheimer* (2023)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

As far back as I can remember, I always wanted to earn a PhD. In fact, my father gave me my name with the meaning of “to give the mission of earning a PhD" when he chose not to pursue one following his master’s degree. What I didn’t anticipate is how hard yet joyful, exhausting yet invigorating, and stressful yet rewarding this journey is. Now, as I finally close this chapter of my life and fulfill my destiny, I want to thank all those who made it possible.

First and foremost, I owe my deepest gratitude to my advisor Prof. Victor Veitch for his unwavering support and invaluable mentorship. Above all, he taught me how to cultivate research taste, and I was fortunate to have the freedom to pursue the ideas I found most compelling. His sharp and insightful comments not only laid the foundation for this thesis but also shaped my academic identity. His vision encouraged me to step beyond my comfort zone and helped me resist my less constructive impulses.

I am also immensely grateful to Prof. Bryon Aragam, with whom I’ve been fortunate to work closely over the years. His meticulousness research approach, exemplary work ethic inspired me and taught me the value of perseverance and attention to detail. His encouragement and special care to mentorship have benefited me immensely—both intellectually and professionally—and his guidance has been instrumental in shaping me as a researcher.

I am deeply thankful to my dissertation committee Prof. Ari Holtzman, and Prof. Yuxin Chen for providing thoughtful insights throughout the process. In particular, Yuxin was the first faculty member I worked with at the University of Chicago, and he helped make this place feel like home.

My research benefited greatly from collaborations with Goutham Rajendran and Prof. Pradeep Ravikumar. Through stimulating discussions and incisive feedback, they turned my scattered ideas into rigorous and coherent research directions.

Throughout this journey, I’ve been incredibly fortunate to forge lasting friendships with

my labmates and colleagues—Zhuokai Zhao, Chaoqi Wang, Zihao Wang, Lin Gui, Jingwen Yang, Kiho Park, Yo Joong Choe, Sean Richardson, Mark Muchane, Zhuorun Chen, David Reber, Irina Cristali, Todd Nief, Andy Arditi, Ming Gao, Chang Deng, Xi Jiang, Zelin Lv, Sheng Qian and many others. Their camaraderie carried me through my lowest moments, inspiring me to press on and grow as a person.

Finally, and by no means least, I owe my deepest gratitude to my family for their unwavering love and encouragement. I dedicate this accomplishment to them.

ABSTRACT

Foundation models, such as large language models (LLMs), operate within vector spaces, whereas human perception of concepts does not naturally align with this framework. This raises a fundamental question: how do these models internalize the structure of concepts within a vector space and how do they use it? To address this, the thesis investigates structural properties such as linearity and partial orthogonality and also studies how models can leverage structures in representations to combine and extract information. The first part analyzes linear representations. While the concept of linearity appears straightforward, its underlying basis—especially in large language models trained solely on next-token prediction—remains a mostly unresolved mystery. This thesis provides new insights into this phenomenon by showing the connection between linear representations and the implicit bias of gradient descent. The second part examines how models represent the intuitive notion of “semantic independence." Rather than formally defining semantic independence, the focus is on the algebraic axioms of independence and how they can be represented in the forms of partial orthogonality in the representation space. Finally, the third part studies representations in a practical setting—fact retrieval—and explores how self-attention can effectively combine stored information in representations to retrieve the most relevant outputs, functioning like associative memory.

CHAPTER 1

INTRODUCTION

Foundation models, such as large language models (LLMs), are on track to becoming a transformative technology of this generation [Bommasani et al., 2021]. Over the past few years, people have figured out how to grow them [Kaplan et al., 2020, OpenAI, 2023] but not how to understand them. Interpreting artificial neural networks shares much in common with neuroscience, but studying foundation models offers distinct advantages. These models provide rich internal signals—such as embeddings and next-token probabilities—and allow for controlled interventions, unlike the human brain. Arguably, developing a theory to understand foundation models may even offer a path toward unlocking the mysteries of neuroscience itself.

At the core of this mission is the quest to make sense of representations. The reason is simple: Because they reflect how models internalize, compress, and reinterpret the world. In other words, representations can be seen as an amalgamation of how the model interprets information from its training data. This gives us a tangible way to infer model's internals. The main focus of this dissertation is on understanding word and token embeddings, though it's worth noting that models may also encode information internally in other ways [Geiger et al., 2021, Geva et al., 2023].

While foundation models operate in vector spaces, humans don't naturally think in those terms. In a sense, these models encode projections of reality within their representations. Borrowing from Plato's allegory of the cave—where prisoners perceive only shadows on a wall—we might view foundation models as the prisoners and their training data as the shadows they observe [Huh et al., 2024]. Why can a finite-dimensional representation capture a reality that may be far more complex? Because it need only encode the relationships among entities—and those relationships often follow simple patterns. For instance, a representation of “cat" doesn't have to store every fact about cats; it only needs to record how “cat" relates to other concepts. This idea dates back to distributional semantics, which underlies modern language models [Firth, 1957]. In this dissertation, we investigate how different structures manifest within representations.

Toward this goal, we face three key challenges. First, intuitive concept structures lack formal mathematical definitions; we must develop formalisms that are both rigorous and intuitively meaningful. Second, foundation models aren't trained to encode such structures—for example, LLMs are trained for next-token prediction—so we need to explain how and why structural patterns emerge naturally. Third, claims about semantic structure are difficult to validate, requiring experimental designs that qualitatively assess our intuitions. This dissertation addresses each of these challenges in turn.

1.1 Outline

This dissertation is organized as follows:

- In Chapter 2, we examine linear representations of concepts defined by pairs of counterfactual tokens. [Park et al., 2023], a particular *geometric structure* in vector-space embeddings. We then explore why such representations arise naturally by training models on next-token prediction tasks and offer a new perspective by relating this phenomenon to the implicit bias of gradient descent. This chapter builds on Jiang et al. [2024b], published in ICML 2024.

- In Chapter 3, we study the notion of “semantic independence", defining it through *algebraic structures* via the abstract independence model [Lauritzen, 1996]. To map this structure into vector spaces, we employ partial orthogonality, which defines an independence model on those spaces. Empirically, partial orthogonality has proven to be a useful tool for analyzing embeddings. This chapter builds on Jiang et al. [2023], published in NeurIPS 2023.

- In Chapter 4, we study how LLMs can combine and extract information from representations of different tokens (i.e., *composite structure*). We focus on fact retrieval, showing that LLMs perform it much like a particular associative-memory model. Through controlled experiments, we further demonstrate how a one-layer transformer—the basic building block of LLMs—can implement associative memory. This chapter builds on Jiang et al. [2024a], published in NeurIPS 2024.

Throughout my PhD, I have also worked on several other projects [Jiang and Veitch, 2022, Jiang and Aragam, 2023, Wang et al., 2023b, Park et al., 2024, Wang et al., 2025], which are omitted from this dissertation due to their indirect relevance. Of these, Park et al. [2024] is the most closely aligned, as it investigates hierarchical and categorical structures in embeddings—paralleling the themes of Chapter 3. Meanwhile, Jiang and Veitch [2022] and Jiang and Aragam [2023] focus on causal representation learning, studying the causal structure of vector space representations, and laying foundation for most of the thesis, which seeks to uncover the semantic structure of embeddings through the lens of latent graphs.

CHAPTER 2
GEOMETRIC STRUCTURE AND LINEAR REPRESENTATIONS

2.1 Introduction

One of the central questions of interpretability research for language models [Mikolov et al., 2013b, OpenAI, 2023, Touvron et al., 2023] is to understand how high-level semantic concepts that are meaningful to humans are encoded in the representations of these models. In this context, a surprising observation is that these concepts are often represented *linearly* [e.g., Mikolov et al., 2013b, Pennington et al., 2014, Arora et al., 2016, Elhage et al., 2022b, Burns et al., 2022, Tigges et al., 2023, Nanda et al., 2023b, Moschella et al., 2022, Park et al., 2023, Li et al., 2023a, Gurnee et al., 2023]. This observation is mainly empirical, leaving open major questions; most importantly: could the apparent "linearity" be illusory?

In this chapter, we study the origins of linear representations in large language models. We introduce a mathematical model for next token prediction in which context sentences and next tokens both reflect latent binary *concept variables*. These latent variables give a formalization of underlying human-interpretable concepts. Using this mathematical model, we prove that these latent concepts are indeed linearly represented in the learned representation space. This result comes in two parts. First, similar to earlier findings on word embeddings [Pennington et al., 2014, Arora et al., 2016, Gittens et al., 2017b, Ethayarajh et al., 2018], we show log-odds matching leads to linear structure. Second, we show that the implicit bias of gradient descent leads to the emergence of linear structure even when this (strong) log-odds matching condition fails. Together, these results provide strong support for the linear representation hypothesis.

There are some noteworthy implications of these results. First, linear representation structure is not specific to the choice of model architecture, but a by-product of how the model learns the conditional probabilities of different contexts and corresponding outputs. Second, the simple latent variable model gives rise to representation behavior such as linearity and orthogonality akin to those observed in LLMs. This suggests it may be a useful tool for further theoretical study in interpretability research.

The development is as follows:

1. In Section 2.2, we present a latent variable model that abstracts the concept dynamics of LLM inference, allowing us to mathematically analyze LLM representations of concepts.

2. Using this model, we show in Section 2.3 that the next token prediction objective (softmax with cross-entropy) and the implicit bias of gradient descent together promote concepts to admit linear representations.

3. A surprising fact about LLMs is that Euclidean geometry on representations sometimes reasonably encodes semantics, despite the Euclidean inner product being unidentified by the standard LLM objective [Park et al., 2023]. In Section 2.4, we show that the implicit bias of gradient descent can result in Euclidean structure having a privileged status such that independent concepts are represented almost orthogonally.

4. Finally, in Section 2.5, we conduct experiments on simulations from the latent variable model confirming that this structure does indeed yield linear representations. Additionally, we assess the theory's predictions using LLaMA-2 Touvron et al. [2023], showing that the simple model yields generalizable predictions.

2.2 Problem setting

In this section, we give a simple latent variable model to study next-token predictions.

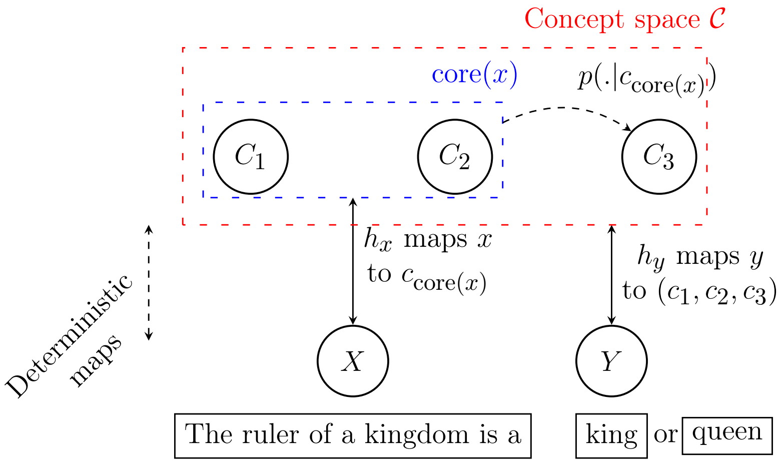

Concept space $\mathcal{C}$
core(x)
$p(.|c_{\text{core}(x)})$
C_1
C_2
C_3
h_x maps x to $c_{\text{core}(x)}$
h_y maps y to (c_1, c_2, c_3)
Deterministic maps
X
Y
The ruler of a kingdom is a
king
or
queen

Figure 2.1: Latent conditional model visualization. An example flow: Suppose X is "The ruler of a kingdom is a". It first maps to a set of core concepts $C_1 = c_1, C_2 = c_2$. This leaves concept $C_3 = \diamond$ unknown, which indicates say "male" or "female". The value of $C_3 = c_3 \in \{0, 1\}$ is determined by a conditional probability $p(.|C_1 = c_1, C_2 = c_2)$ and once it's determined, the entire $c = (c_1, c_2, c_3)$ can be mapped to the next token Y, e.g. "king" (if $c_3 = 0$) or "queen" (if $c_3 = 1$) respectively.

2.2.1 Latent conditional model

We let context sentences and next tokens share the same latent space where the *probabilistic inference* happens. This lets us transform the next token prediction problem into the problem of learning various conditional probabilities among the latent variables. For a visualization of the framework, see Figure 2.1.

Latent space of concepts We model the latent space with a set of binary random variables: $V_C = \{C_1, ..., C_m\}$. Intuitively, one can view each latent binary random variable as representing the existence of a *concept*, e.g. positive vs negative sentiment, and V_C contains all relevant concepts of interest for language modeling. Notation-wise, we will use $C = (C_1, \ldots, C_m)$ to represent the latent random vector and $c \in \{0, 1\}^m$ to denote realizations of the latent random vector. For a index set $I \subseteq [m]$, C_I (similarly, c_I) denotes a subset of random variables. Note that the latent sample space $\mathcal{C}$ is the collection of binary vectors $\{0, 1\}^m$.

These binary random variables are not necessarily independent. To model their dependencies, we assume that the variables form a Markov random field. That is, there is an undirected graph $G_C = (V_C, E_C)$ such that $p(C_i|C_{[m]\setminus i}) = p(C_i|C_{\mathrm{ne}(i)})$ for all $i \in [m]$ (where $\mathrm{ne}(i)$ denotes the set of neighbors of i), i.e. a variable is conditionally independent of all other variables given its neighbors. Markov random fields capture a wide variety of probability distributions, which include independent random variables as a special case.

Remark 1. The theory readily handles directed acyclic graphical models (DAGs) [Koller and Friedman, 2009], but we work with undirected graphs for notational simplicity.

Latent space to next token (y) Let $Y \in \mathcal{Y}$ be the random variable denoting the next token with sample space $\mathcal{Y}$. We assume there exists an injective measurable function from concept space $\mathcal{C}$ to token space $\mathcal{Y}$. That is, the value of C can always be read off from Y. This map induces a distribution on the next token (via the pushforward measure) that the LLM tries to learn. We denote the inverse of this map as h_y (mapping tokens to concepts). The injectivity assumption is made for simplicity—with the injectivity assumption, for any given concept, one can identify pairs of tokens that only differ in that concept. This is useful for the theoretical analysis.

Context sentence (x) to latent space Let $X \in \mathcal{X}$ be the random variable denoting context sentence where $\mathcal{X}$ is the sample space, e.g. "The ruler of a kingdom is a" is an element of $\mathcal{X}$. Intuitively, X contains partial information about the latent concepts that determine the next token Y. This is similar to a hidden Markov model where one can use previous observations to infer the current latent state. In other words, if one models language generation as transitions/drifts in the latent space [Arora et al., 2016], then given the context, one cannot fully determine but can make educated guesses on where the latent space might be transitioned into. Note that there's inherent ambiguity involving the next token prediction, e.g., multiple tokens could follow the sentence "I am a ".

Now, we will describe precisely how contexts X and concepts C relate to each other.

First, observe that not every concept combination can induce the context sentence of interest, e.g., not every human-interpretable concept is associated with the sentence “The ruler of a kingdom is a”. To capture the notion of concepts relevant to a specific prompt x we define:

$$\mathcal{C}^x = \{c \in \mathcal{C} = \{0,1\}^m : P(x|C = c) \neq 0\} \tag{2.2.1}$$

$$\text{core}(x) = \{i \in [m] : (c_i = 1, \forall c \in \mathcal{C}^x) \text{ or } (c_i = 0, \forall c \in \mathcal{C}^x)\}. \tag{2.2.2}$$

Here, $\mathcal{C}^x$ is the set of all realizations of latent concept vectors in $\mathcal{C} = \{0,1\}^m$ that can induce x. In other words, $\mathcal{C}^x$ contains all concept vectors c that are related to x. Within this subset, core(x) indicates the concepts that are always “on” or “off”. See Figure 2.1 for an illustration. In other words, given x, the values of “core concepts” are fixed and the rest of the non-core concepts are left to be determined. This represents the determinacy in the relationship between X and C. To capture this determinancy relationship we define the map $h_x : \mathcal{X} \to \{\diamond, 0, 1\}^m$ (where $\diamond$ stands for “unknown” and is to be read as “unknown”) as

$$h_x(x)_i = \begin{cases} c_i & \text{if } i \in \text{core}(x) \text{ and } c \in \mathcal{C}^x \\ \diamond & \text{if } i \notin \text{core}(x) \end{cases}$$

where c_i is the known quantity for the core concept of index i. Let’s denote $\mathcal{D} = \{\diamond, 0, 1\}^m$, the set of all possible conditioning contexts.

Predictive distribution We make the modeling choice $p(c|x) = p(c|c_{\text{core}(x)})$—the relationship between x to c factorizes via the core concepts. This captures the idea that only the core concepts present in a context sentence are relevant for the next token prediction. So, e.g., the sentence “The ruler of a kingdom is a ” has an induced probability of the gender concept. This induced probability is determined by the conditional prob of core concepts (e.g. `is_royalty` $= 1$) of the sentence. In other words, given a context sentence x, the joint posterior distribution of latent concepts outside of "core concepts"—which are fixed by x—is determined by the latent space. Thus, we have

$$p(y|x) = p(c|x) = p(c|c_{\text{core}(x)})$$

where $c = h_y(y)$ and the first equality follows from injectivity.

Notation We summarize our notation here.

- $\mathcal{X}$ – Set of all context sentences
- $\mathcal{Y}$ – Set of all tokens
- $\mathcal{C} = \{0, 1\}^m$ – The set of binary concept vectors
- $\mathcal{D} = \{\diamond, 0, 1\}^m$ – The set of context vectors with some unknown concepts
- $\text{core}(x) \subseteq [m]$ – The core concepts in x
- h_x – Deterministic map from X to $\mathcal{D}$
- h_y – Deterministic map from Y to $\mathcal{C}$

2.2.2 Next token prediction

The goal of the next token prediction is to learn $p(y|x)$. Such probabilities are outputs by autoregressive large language models. In particular, LLMs learn two functions, the embedding $\bar{f} : \mathcal{X} \to \mathbb{R}^d$ and the unembedding $\bar{g} : \mathcal{Y} \to \mathbb{R}^d$ such that

$$p(y|x) \approx \hat{p}(y|x) = \frac{\exp(\bar{f}(x)^T \bar{g}(y))}{\sum_y \exp(\bar{f}(x)^T \bar{g}(y))}$$

Note that under our model, $p(y|x) = p(c|c_{\text{core}(x)})$. Therefore, we can assume $\bar{f}, \bar{g}$ depends on the latents c as well through the functions f, g. In particular, define $\bar{f} = f \circ h_x$ and $\bar{g} = g \circ h_y$, i.e. $\bar{f}(x) = f(h_x(x))$ and $\bar{g}(y) = g(h_y(y))$.

Recall that $\mathcal{C} = \{0,1\}^m$ denotes the collection of all binary concept vectors, and $\mathcal{D} = \{\diamond, 0, 1\}^m$ is the set encompassing all conditions. Then, equivalently, under the latent conditional model, f and g are trained such that their inner product can be used to estimate various conditional probabilities of the following form,

$$p(c|d) \approx \hat{p}(c|d) = \text{softmax}(f(d)^T g(c))$$

for all $c \in \widehat{\mathcal{C}}$ and $d \in \widehat{\mathcal{D}}$ where $\widehat{\mathcal{C}} \subseteq \mathcal{C}$ and $\widehat{\mathcal{D}} \subseteq \mathcal{D}$. $\widehat{\mathcal{C}}$ and $\widehat{\mathcal{D}}$ represent the binary vectors and contexts present in the training dataset. To prove the full slate of our results, unless otherwise stated, we assume $\widehat{\mathcal{C}} = \mathcal{C}$ and $\widehat{\mathcal{D}} = \mathcal{D}$. We will show in Section 2.5 what happens when this does not hold.

In our analysis, without loss of generality, we represent elements of $\widehat{\mathcal{C}}$ and $\widehat{\mathcal{D}}$ as one-hot encodings. We let f and g assign unique vectors to each element (i.e., the functions are embedding lookups).

2.3 Linearity

In this section, we study the phenomenon of linear representations, which we now formally define. Adapting [Park et al., 2023], we introduce the following definition within the latent conditional model. Recall that the cone of a vector v is defined as $\text{Cone}(v) = \{\alpha v : \alpha > 0\}$. For a concept $c \in \mathcal{C}$ and $t \in \{0,1\}$, let $c_{(i \to t)}$ indicate the concept c with the ith concept set to t, i.e. the jth concept of $c_{(i \to t)}$ is c_j if $j \neq i$ and t otherwise. A similar notation can be defined for vectors in $\mathcal{D}$ as well. In addition, we refer to vectors that only differ in one latent concept as counterfactual pairs (e.g. $c_{(i \to 1)}, c_{(i \to 0)}$) and the differences between their representations as steering vectors (e.g. $g(c_{(i\to 1)}) - g(c_{(i\to 0)})$).

Definition 2 (Linearly encoded representation)**.** A latent concept C_i is said to have *linearly encoded representation in the unembedding space* if there exists a unit vector u such that $g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \in \mathrm{Cone}(u)$ for all $c \in \widehat{\mathcal{C}}$. Similarly, we say that C_i has a *linearly encoded representation in the embedding space* if there exists a unit vector v such that $f(d_{(i\to 1)}) - f(d_{(i\to 0)}) \in \mathrm{Cone}(v)$ for all $d \in \widehat{\mathcal{D}}$. In addition, we say C_i has *a matched*-representation if $u = v$.

We first prove that linearly encoded representations will arise in a subspace as a result of log-odds matching (Section 2.3.1) which relies on assumptions on the training data distributions. However, empirical observations (Section 2.5) reveal that, within the latent conditional model, linear representations in both the embedding and unembedding space can emerge without depending on the underlying graphical structures or the true conditional probabilities. This leads us to establish a connection between the linearity phenomenon and the implicit bias of gradient descent on exponential loss in Section 2.3.2, which forms the main technical contribution of our work.

2.3.1 Linearity from log-odds

In this section, we show that if the log odds of the learned probabilities is a constant that depends only on the concept of interest, then we must have linearity of representations for that concept in a subspace. This result is in line with many existing works on word embeddings [Pennington et al., 2014, Arora et al., 2016, Gittens et al., 2017a, Ethayarajh et al., 2018, Allen et al., 2019, Ri et al., 2023]. This shows that the latent conditional model is a viable theoretical framework for studying LLMs. Going beyond prior works, our findings additionally connect linearity with the graphical structure of the latent space. An illustrative example is the case of independent concepts, discussed below.

Concretely, the goal of this section is to argue that for any concept $i \in [m]$, the steering vectors in the unembedding space $\Delta_{c,i} = g(c_{(i \to 1)}) - g(c_{(i \to 0)})$ will all be parallel for all $c \in \mathcal{C}$ (up to an ambiguous subspace we cannot control).

Before we state the main theorem, we motivate the assumptions. Firstly, we can reasonably assume the log-odds condition, which states that the model has been trained well enough to have learned correct log-odds for any concept i, under all contexts not conditioning on i. Secondly, since we cannot take the logarithm of 0 conditional probabilities, it's difficult to predict how the steering vector behaves in the directions $f(d)$ for $d_i \neq \diamond$. Therefore, in this section, we project out this ambiguous subspace for now. However, as we will show in Section 2.3.2, learning these 0 conditional probabilities with gradient descent promotes linear representations overall.

As motivated above, define $\overline{\Delta_{c,i}} = \Pi_i \Delta_{c,i}$ where Π_i is the projection operator onto the space $\text{span}\{f(d) | d_i = \diamond\}$, i.e., we project onto the space of contexts that does not condition on C_i.

Theorem 3 (Log-odds implies linearity)**.** *Fix a concept $i \in [m]$. Suppose for any concept vector $c \in \mathcal{C}$ and context $d \in \mathcal{D}$ such that $d_i = \diamond$, we have*

$$\ln \frac{\hat{p}(c_{(i \to 0)} | d)}{\hat{p}(c_{(i \to 1)} | d)} = \ln \frac{p(C_i = 0)}{p(C_i = 1)}$$

Then, the vectors $\overline{\Delta_{c,i}}$ over all $c \in \mathcal{C}$ are parallel.

The proof is deferred to Appendix A.1. This theorem states that for a fixed concept C_i, all concept steering vectors are parallel to each other, regardless of the other concepts (if we project out the ambiguous subspace). This can help to explain that in the representation space, one can have $\bar{g}(\text{"king"}) - \bar{g}(\text{"queen"})$ (approximately) parallel to $\bar{g}(\text{"man"}) - \bar{g}(\text{"woman"})$ [Mikolov et al., 2013b, Park et al., 2023]. In this case, the two pairs ("king", "queen") and ("man", "woman") only differ in the gender concept.

The case of independent concepts We now justify that the assumption in Theorem 3 holds when the concepts are jointly independent in the training distributions, therefore the theorem applies to such concepts as a special case. Indeed, in the jointly independent case, the true distribution p is a product distribution, therefore the expression on the left-hand side must match the right-hand side above and the assumption holds.

In summary, Theorem 3 applies to the case of jointly independent concepts (and also more generally), implying that matching log-odds formally implies linearity of concept representations in a subspace. In Appendix A.2, we further generalize this result to the case when the concepts are not necessarily independent and instead come from an MRF (or a DAG), where we show using log-odds that concept representations lie in a subspace of small dimension. However, the assumption of matching log-odds may be restrictive. In order to go beyond, we invoke ideas from the optimization literature on the implicit bias of gradient descent, which we outline in the next section.

2.3.2 Linearity from implicit bias of gradient descent

The goal of this section is to argue that implicit bias of gradient descent also promotes linearity in the entire space of representations. We have seen in the previous section that log-odds matching can lead to linearity. However, in the context of language modeling, it is stringent to require concepts to have matching log-odds under every possible conditioning. Furthermore, the peculiar case of linearity in the latent conditional model is that the phenomenon can occur even for randomly generated graphs and parameters of the underlying distributions (Section 2.5). This raises the question of whether there are other factors influencing the representation of concepts.

We now study the role of gradient descent in this phenomenon. However, because the next token prediction is a highly complex nonconvex optimization problem, rather than delving into the entire optimization process, our focus is on identifying *subproblems* within this optimization that can result in linearly encoded representations. The goal here is to highlight the underlying dynamics that prefer linear representations. It is worth noting that we choose to study the role of gradient descent instead of the more computational tractable stochastic gradient descent used in experiments for analytic simplicity. However, according to classical stochastic approximation theory [Kushner and Yin, 2003], with a sufficiently small learning rate, the additional stochasticity is negligible, and stochastic gradient descent will behave highly similarly to gradient descent [Wu et al., 2020].

The key observation is that there is a hidden binary classification task. Let's consider the simple three-variable example. Suppose instead of predicting all possible conditional distributions of latents, the model only learns $p(\cdot|c_1 = 1)$ and $p(\cdot|c_1 = 0)$. Denoting $v = g(1, 1, 1) - g(0, 1, 1), w_0 = f(0, \diamond, \diamond), w_1 = f(1, \diamond, \diamond)$,

$$\frac{\hat{p}(0, 1, 1|c_1 = 1)}{\hat{p}(1, 1, 1|c_1 = 1)} = \exp(-w_1^T v) \approx 0,$$
$$\frac{\hat{p}(1, 1, 1|c_1 = 0)}{\hat{p}(0, 1, 1|c_1 = 0)} = \exp(w_0^T v) \approx 0$$

Equivalent, the model is trying to optimize the following loss:

$$L(w_1, w_0, v) = \sum_{i=0}^{1} \ell(-y_i w_i^T v)$$

where $y_0 = 1, y_1 = -1$ and $\ell(x) = \exp(-x)$. Intuitively, if w_0 and w_1 are fixed, then this is a binary classification task with exponential loss. It is known that gradient descent under this setting would converge to the max-margin solution [Soudry et al., 2018], which makes the direction of v unique.

The following theorem makes this intuition precise. It states that by optimizing a specific subproblem of the next token predictions using gradient descent with *embeddings fixed*, latent unembedding representations will be encoded linearly. For vectors u, v, denote $\cos(u, v) =$

$\frac{\langle u,v \rangle}{||u|| \cdot ||v||}$.

Theorem 4 (Gradient descent with fixed embeddings)**.** *Fix $i \in [m]$. Let $\widehat{\mathcal{D}} = \{d^{\diamond}_{(i\to 1)}, d^{\diamond}_{(i\to 0)}\}$ where $d^{\diamond} = [\diamond, ..., \diamond]$ and $\Delta_{c,i} = g(c_{(i\to 1)}) - g(c_{(i\to 0)})$. Suppose the loss function is the following:*

$$L(\{\Delta_{c,i}\}_c, f(d^{\diamond}_{(i\to 1)}), f(d^{\diamond}_{(i\to 0)})) = \sum_{c \in \overline{\mathcal{C}}} \Big(\exp(-\Delta^T_{c,i} f(d^{\diamond}_{(i\to 1)})) + \exp(\Delta^T_{c,i} f(d^{\diamond}_{(i\to 0)})) \Big)$$

where for all $c \in \overline{\mathcal{C}}$, $c_i = 1$ and $f(d^{\diamond}_{(i\to 1)})) \neq f(d^{\diamond}_{(i\to 0)})$. Then fixing f and training g using gradient descent with the appropriate step size, we have

$$\lim_{t\to\infty} \cos(\Delta^t_{c^1,i}, \Delta^t_{c^2,i}) = 1$$

for any $c^1, c^2 \in \overline{\mathcal{C}}$ where the superscript t is meant to represent vectors after t number of iterations.

All proofs are deferred to Appendix A.3. The theorem says that under gradient descent with the exponential loss function and embeddings fixed, the unembedding vector differences for a fixed concept are all aligned in the limit. It is worth mentioning that although the loss function is a bit simplified, it does represent the significant subproblem of the optimization in terms of linear geometry. The reasons are two-fold: (1) Because conditional probabilities are learned with the softmax function, there is an extra degree of freedom. In other words, one does not need to use all the logits to estimate distributions, just the differences between them. (2) Due to the nature of the exponential function, the closer the ratios between conditional distributions are to zero, the larger the distances between unembeddings need to be. Therefore, learning to predict these zero conditional probabilities would dominate the direction of concept representations.

The above theorem assumes that the embedding space is fixed. It turns out that gradient descent will also have the tendency to align embedding and unembedding space when they're not fixed and are trained jointly, as we will show next. First, one can observe that if the 0 conditional probabilities are learned to a reasonable approximation, then the inner product of embedding and unembedding steering vectors of the same concept should be large (Proposition 31) or in other words, the exponential of the inner product is approaching infinity. But since there are different ways that this inner product can reach infinity, a natural question to ask is what happens when one purely optimizes this exponential learning objective. Theorem 32 shows that gradient descent, when trained to minimize the exponential of negative inner product of two vectors, would align these two vectors.

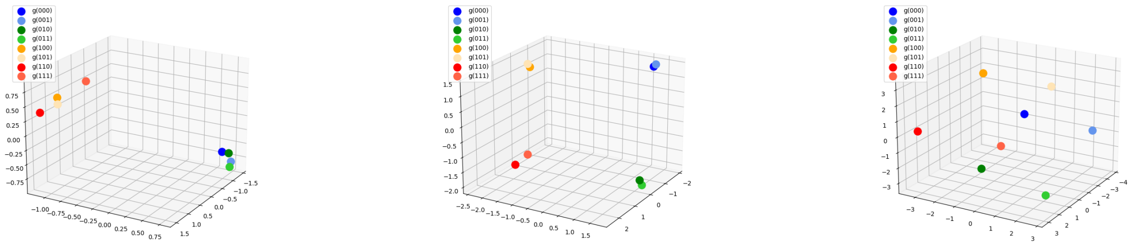
Finally, using Theorem 4 and Theorem 32, we obtain the following statement, which is one of our core contributions.

Theorem 5 (Gradient descent aligns representations). *Under the conditions of Theorem 4, if one further assumes that $\{\Delta_{c,i}\}_C, f(d^\diamond_{(i\to 1)})), f(d^\diamond_{(i\to 0)})$ are initialized to be mutually orthogonal with the same norm, then optimizing f and g using gradient descent, we have*

$$\lim_{t\to\infty} \cos(\Delta^t_{c^1,i}, \Delta^t_{c^2,i}) = 1$$
$$\lim_{t\to\infty} \cos(\Delta^t_{c^1,i}, f^t(d^\diamond_{(i\to 1)}) - f^t(d^\diamond_{(i\to 0)})) = 1$$

for any $c^1, c^2 \in \overline{\mathcal{C}}$ where the superscript t is meant to represent vectors after t number of iterations.

While stated generally as a result of the properties of gradient descent, Theorem 5 says that when the latent conditional models are trained on a subproblem of next token prediction with reasonable initializations (see below remark), implicit bias of gradient descent will align the embedding and unembedding vectors as well as aligning among the unembedding vectors. This can manifest as linear representations in the final trained LLMs such as LLaMA-2 [Touvron et al., 2023]. Although the theorem addresses only a specific subproblem, it no-



(a) $\widehat{\mathcal{D}} = \{(0, \diamond, \diamond), (1, \diamond, \diamond)\}$ **(b)** $\widehat{\mathcal{D}} = \{(0, 0, \diamond), (0, 1, \diamond), (1, 0, \diamond), (1, 1, \diamond)\}$ **(c)** $\widehat{\mathcal{D}} = \mathcal{D}$

Figure 2.2: Unembedding representations form different clusters after training on various conditioning sets. This figure shows unembedding representations after learning different sets of conditional distributions.

tably suggests, as discussed earlier, the underlying bias and dynamics that contribute to the emergence of linear representations. That is, if one solely optimizes this subproblem, one can get perfect linear representations. We verify this result empirically in Section 2.5.

Remark 6. The initial condition assumption in the theorem statement seems unnecessary as evidenced by our experiments in Section 2.5. On the other hand, if one initializes vectors with high dimensional multivariate Gaussians, then the assumption will be approximately satisfied with high probability [Dasgupta, 1999].

The role of learning various conditional distributions So far, we have only discussed learning conditional distributions for a single concept. In general, due to the presence of zero probabilities in various conditioning contexts, learning these conditional probabilities is, in some sense, learning different forces to push sets of representations far apart because one needs relatively large inner products to get exponents to be close to zero.

Figure 2.2 gives an illustrative example with three concept variables that shows how learning different subsets of conditional probabilities would push various subsets of unembeddings in different ways. For example, if the model exclusively learns conditional probabilities for C_1, the unembedding representations would form clusters based on the values of C_1. Similarly, if the model learns conditional probabilities for both C_1 and C_2, the unembedding representations would cluster based on the values of both C_1 and C_2. However, in the second case, the concept directions of C_1 and C_2 are coupled. As shown in Figure 2.2(b), $g(110) - g(100)$ is parallel with $g(111) - g(101)$ but not with $g(010) - g(000)$. To decouple them, one needs a bigger $\widehat{\mathcal{D}}$ as in Figure 2.2(c). That said, it seems that a complete set of $\mathcal{D}$ is not always necessary (Section 2.5).

An analogy would be to view each binary vector as a combination of "electrons" where the values represent positive or negative charges. Learning probabilities under different conditioning contexts can be likened to applying various forces to manipulate the positions of these electrons. This is similar to the Thomson Problem in chemistry and occurs in other contexts of representation learning as well [Elhage et al., 2022b].

2.4 Orthogonality

A surprising phenomenon observed in LLMs like LLaMA-2 [Touvron et al., 2023] is that the Euclidean geometry can somewhat capture semantic structures, despite the fact that the Euclidean inner product is not identified by the training objective [Park et al., 2023]. Consequently, one notable outcome is the tendency for semantically unrelated concepts to be represented as almost orthogonal vectors in the unembedding space. In this section, we study this phenomenon under our latent conditional model. Based on our underlying Markov random field distribution, we define unrelated concepts to be those separated in G_C.

Theorem 7. *Let $\widehat{\mathcal{C}} = \mathcal{C}$ and $\widehat{\mathcal{D}} = \mathcal{D}$. Assuming $p(c) > 0$ for any $c \in \mathcal{C}$ and C_i and C_j are two latent variables separated in G_C. Given any binary vector $c \in \mathcal{C}$, there exists a subset $\mathcal{D}_c \subset \mathcal{D}$ such that $d_i = \diamond$ and $p(c|d) > 0$ for any $d \in \mathcal{D}_c$. If one further assume that $\hat{p}(c|d) = p(c|d)$ for any $d \in \mathcal{D}_c$, then*

$$g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \perp f(d_{(j\to c_j)}) - f(d_{(j\to\diamond)})$$

for any $d \in \mathcal{D}_c$.

All proofs are deferred to Appendix A.4. As suggested in Section 2.3.2 and verified empirically in Section 2.5, under the latent conditional model, representations are not only linearly encoded but also frequently aligned between embedding and unembedding spaces, which implies the following corollary on the orthogonal representation of unrelated concepts.

Corollary 8. *Under the conditions of Theorem 7, if concept C_i and C_j have matched-representations, then*

$$g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \perp g(c_{(j\to 1)}) - g(c_{(j\to 0)})$$

for any $c \in \mathcal{C}$.

In simulation, given enough dimensions, the latent conditional model tends to learn near orthogonal representations regardless of graphical dependencies. One plausible rationale behind this phenomenon is the tendency for steering vectors in both unembedding and embedding spaces to possess large norms because, by Proposition 31, the inner product of embedding and unembedding steering vectors of the same concept is large. However, for steering vectors representing distinct concepts, it's often necessary for the inner product to be small as dictated by what's given in the training distributions. To accommodate for this, their cosine similarities tend to be close to zero. As a consequence, unembedding representations of different concepts often end up being near orthogonal as well.

Remark 9. It is worth noting that the conditions for orthogonal representations presented here can break in practice for LLMs. For instance, the injectivity assumption might not hold, and embedding and unembedding representations might not have perfectly matched representations.

2.5 Experiments

In this section, we present two slates of experiments to validate and augment our theoretical contributions. Specifically, in Section 2.5.1, we run experiments on simulated data from the latent conditional model to verify the existence of linear and orthogonal representations in this theoretical framework and how training on smaller dimensions, incomplete sets of contexts, and concept vectors can affect the representations. Then, we run experiments on large language models in Section 2.5.2 to show nontrivial alignment between embedding and unembedding representations of the same concept as predicted by our theory.

2.5.1 Simulated experiments

Simulation setup For simulation experiments, we first create simulated datasets by initially creating random DAGs (see Remark 1) with m variables/concepts. For a specific random DAG, the conditional probabilities of one variable given its parents are modeled by Bernoulli distributions where the parameters are sampled uniformly from $[0.3, 0.7]$. For example, suppose the DAG is $C_1 \to C_2$. Then $C_1 \sim \text{Bern}(p_1)$, $(C_2|C_1 = 0) \sim \text{Bern}(p_2)$, $(C_2|C_1 = 1) \sim \text{Bern}(p_3)$ where $p_1, p_2, p_3 \sim \text{Unif}\left([0.3, 0.7]\right)$. In other words, we generate random graphical models wherein both the structures and distributions are created randomly. From these random graphical models, we can sample values of variables which are binary vectors as our datasets.

To let models learn conditional distributions, we train them to make predictions under cross-entropy loss after randomly masking the sampled binary vectors. Unless otherwise stated, the model is trained using stochastic gradient descent with a learning rate of 0.1 and batch size 100. More details are deferred to Appendix A.5.

Complete set of conditionals ($\widehat{\mathcal{C}} = \mathcal{C}, \widehat{\mathcal{D}} = \mathcal{D}$) We first run experiments where the model is trained to learn the complete set of conditional distributions. The representation

Table 2.1: When the model is trained to learn the complete set of conditionals, the m latent variables are represented linearly, and the embedding and unembedding representations are matched. The table shows average cosine similarities among and between steering vectors of unembeddings and embeddings. Standard errors are over 100 runs for 3 variables and 4 variables, $50, 20$ and 10 runs for $5, 6$ and 7 variables respectively.

m	Unembedding	Embedding	Unembedding and Embedding
3	0.972±0.006	0.982±0.005	0.980±0.005
4	0.975±0.005	0.971±0.005	0.973±0.005
5	0.988±0.004	0.981±0.004	0.984±0.004
6	0.997±0.000	0.985±0.002	0.990±0.001
7	0.995±0.001	0.972±0.004	0.981±0.003

dimension is set to be the same as the number of latent variables. Given a concept, to measure linearity, we calculate the average cosine similarities among steering vectors in the unembedding space and in the embedding space as well as between steering vectors in these two spaces. The result is shown in Table 2.1 which suggests that under the latent conditional model with the complete set of contexts, learned representations are indeed linearly encoded. In addition, Figure A.1 in Appendix A.5 shows that as loss decreases, the cosine similarities increase rapidly. Due to the exponential number of vectors that need to be trained, one cannot directly simulate a large number of latent variables. However, as we will show later (Appendix A.5), one can use incomplete sets $\widehat{\mathcal{C}}$ and $\widehat{\mathcal{D}}$ and simulate with more latent variables.

Orthogonality Section 2.4 argues that, under the latent conditional model, unembedding representations of separated concepts will be orthogonal. In practice, Figure A.3(a) (Appendix A.5) shows that this happens for simulated data even without latent variables being separated in the latent graph. We test the orthogonalities of representations in the latent conditional model by calculating the average cosine similarities between different sets of steering vectors. To see how well the theoretical framework fits LLMs, we similarly calculate the average cosine similarities between different sets of steering vectors in LLaMA-2

[Touvron et al., 2023] using the counterfactual pairs in [Park et al., 2023] and they're reported in Figure 2.3. In particular, Figure A.3 (Appendix A.5) shows that both simulated experiments and LLaMA-2 experiments exhibit similar behaviors where steering vectors of the same concept are more aligned while steering vectors of different concepts are almost orthogonal. This validates the claim that the latent conditional model is a suitable proxy to study LLMs.

In real-life datasets, not every concept or context vector has a natural correspondence to a token or sentence. On the other hand, the number of tokens and sentences grows exponentially with the number of latent concepts. Therefore, it is not realistic to always have $\widehat{\mathcal{D}} = \mathcal{D}$ or $\widehat{\mathcal{C}} = \mathcal{C}$. In other words, one does not need the next token prediction to learn all possible conditional probabilities perfectly. Because we are in a simulated setup, it's easy to probe the behavior in these situations. Therefore, we also perform additional experiments to observe these aspects of our simulated setup, in particular (i) Training on an incomplete set of contexts ($\widehat{\mathcal{D}} \subset \mathcal{D}$), (ii) Incomplete set of concept vectors ($\widehat{\mathcal{C}} \subset \mathcal{C}$). These are deferred to Appendix A.5. The experiments show that linearity is robust to these changes.

Moreover, in practice, the representation dimension is typically much smaller than the number of concepts represented. Thus, we run additional experiments in Appendix A.5 with decreasing dimensions and observe one can still get reasonable linear representations.

Finally, we show in Appendix A.5 that gradient descent or stochastic gradient descent is not the only algorithm that can induce linear representation, running other first-order methods like Adam [Kingma and Ba, 2015] can lead to a similar pattern.

2.5.2 Experiments with large language models

In this section, we will conduct experiments on pre-trained large language models. We first emphasize that prior works [Elhage et al., 2022b, Burns et al., 2022, Tigges et al., 2023, Nanda et al., 2023b, Moschella et al., 2022, Park et al., 2023, Gurnee et al., 2023]

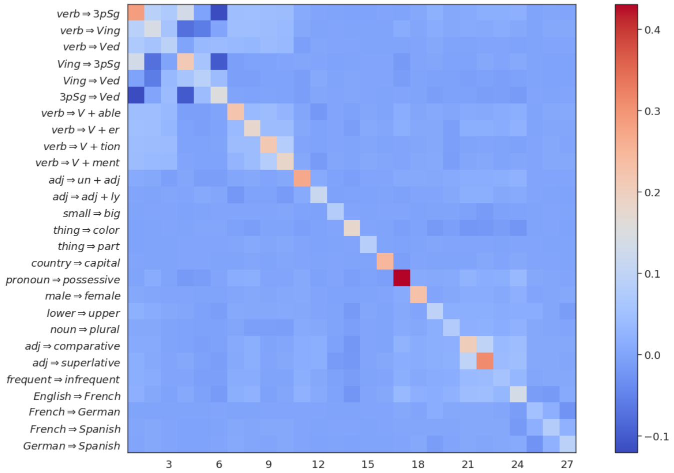


Figure 2.3: Unembedding steering vectors of the same concept in LLaMA-2 have nontrivial alignment, but steering vectors of different concepts are represented almost orthogonally.

have already exhibited certain geometries of LLM representations related to linearity and orthogonality, via experiments. Therefore, in this section, we probe the geometry of the LLM representations, in particular that of the interplay between the embeddings and unmbeddings of the contexts and the tokens, that have not been explored in prior works. As Park et al. [2023] remark, for a given binary concept, it's hard to generate pairs of contexts that differ in precisely this context, partly because of nuances of natural language and mainly because such counterfactual sentences are hard to construct even for human beings. In this section, we try to recreate these sets of missing experiments in the literature using existing open-source datasets.

Multilingual embedding geometry We first consider language translation concepts. For the embedding vectors, we consider pairs of contexts (x^0, x^1) where x^0, x^1 are the same sentences but in different languages. We consider four language pairs French–Spanish, French–German, English–French, and German–Spanish from the OPUS Books dataset [Tiedemann, 2012]. We take 150 random samples and filter out contexts with less than 20 or more than 150 tokens. For the unembedding concept vectors, we use the 27 concepts as described in Park et al. [2023], which were built on top of the Big Analogy Test dataset [Gladkova et al., 2016]. Examples of both datasets and a list of the 27 concepts are shown in Appendix A.6.

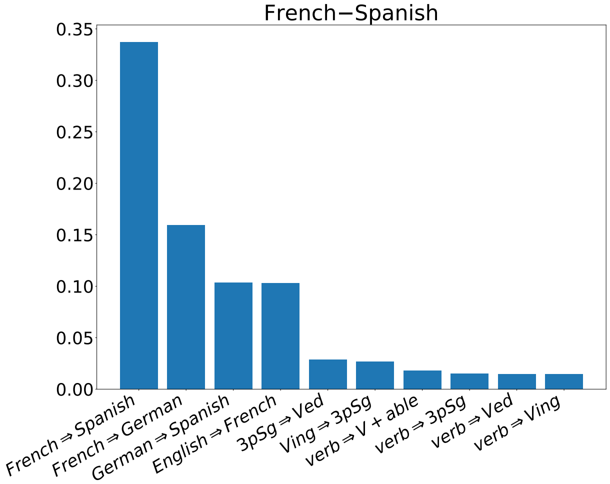


Figure 2.4: The French–Spanish concept is highly correlated with similar token concepts relative to others. Figure shows cosine similarities between the French–Spanish concept and token concepts.

We consider embeddings and unembeddings from LLaMA-2-7B [Touvron et al., 2023]. We compute the absolute cosine similarity between the average embedding steering vector and the average unembedding steering vector. A barplot for French–Spanish with top 10 similarities is shown in Figure 2.4. The entire barplot with 27 concepts and the barplots for the other language translation concepts are in Appendix A.6. As we can see, there is alignment between the embedding and unembedding representations for matching concepts relative to unmatched concepts, as Theorem 5 predicts.

Winograd Schema Next, we consider counterfactual context pairs arising from the Winograd Schema dataset [Levesque et al., 2012], which is a dataset of pairs of sentences that differ in only one or two words and which further contain an ambiguity that can only be resolved with world knowledge and reasoning. We run experiments similar to the multilingual embedding geometry experiments and also run additional experiments on similarities between Winograd context pairs and the 27 concepts from Park et al. [2023]. The experiment details are in Appendix A.6.

These LLM experiments show that matching contexts are better aligned with the corresponding unembedding steering vectors than non-matching contexts, as predicted from our theory. Note that although the final cosine similarities are lower than what was exhibited in the simulated experiments, this is not surprising due to the complexity and nuances of natural language and LLMs. Another reason is that abstract concepts do not necessarily lie in a one-dimensional space, so standard cosine similarity metrics may not be an optimal choice here. However, the partial alignment already serves to strongly validate our theoretical insights.

2.6 Related Work

Linear representations Linear representations have been observed empirically in word embeddings [Mikolov et al., 2013b, Pennington et al., 2014, Arora et al., 2016] and large language models [Elhage et al., 2022b, Burns et al., 2022, Tigges et al., 2023, Nanda et al., 2023b, Moschella et al., 2022, Park et al., 2023, Gurnee et al., 2023]. Many works in the pre-LLM era, also attempt to explain this phenomenon theoretically. For instance, Arora et al. [2015] and their follow-up works [Arora et al., 2018, Frandsen and Ge, 2019] study the RAND-WALK model in which latent vectors undergo continuous drift on the unit sphere. A similar notion can be found also in dynamic topic modeling [Blei and Lafferty, 2006] and its subsequent works [Rudolph et al., 2016, Rudolph and Blei, 2017]. In contrast, the latent conditional model in this chapter studies discrete latent concept variables that can capture semantic meanings.

Other works include the paraphrasing model [Gittens et al., 2017a, Allen and Hospedales, 2019, Allen et al., 2019], which proposes to study a subset of words that are semantically equivalent to a single word, however the uniformity assumption is somewhat unrealistic. Ethayarajh et al. [2018] explore how the linearity property can arise by decomposing point-wise mutual information matrix and Ri et al. [2023] examine the phenomenon from the perspective of contrastive loss which all rely on assumptions on the matching of probability ratios.

Wang et al. [2023d] also use a latent variable model between prompt and output to establish the linear structure of representations. However, they study this phenomena in the context of text-to-image diffusion models, and their construction relies heavily on the Stein score representation. By contrast, the model here is closer to the standard decoder-only LLM setup, and highlights the key role of softmax.

Linearity has also been observed in other domains such as computer vision [Radford et al., 2015, Raghu et al., 2017, Bau et al., 2017, Engel et al., 2017, Kim et al., 2018, Trager et al., 2023, Wang et al., 2023d] and other intelligent systems [McGrath et al., 2022, Schut et al., 2023]. We expect our ideas to extend to other domains, however we restrict our attention to language modeling in this work.

Geometry of representations There's a body of work on studying the geometry of word and sentence representations [Mimno and Thompson, 2017a, Reif et al., 2019, Volpi and Malagò, 2021, 2020, Li et al., 2020, Chen et al., 2021, Chang et al., 2022, Liang et al., 2022, Jiang et al., 2023, Park et al., 2023]. In particular, Mimno and Thompson [2017a] and Liang et al. [2022] discover representation gaps in the context of word embeddings and vision-language models. Park et al. [2023] attempts to define a suitable inner product that captures semantic meanings. Jiang et al. [2023] study the connection between independence and orthogonal representation by adopting the abstract notion of independence models. One can also view embeddings in the context of information geometry [Volpi and Malagò, 2021, 2020].

Causal representation learning There is a subtle connection between our work and the field of causal representation learning that we now briefly comment on. Causal representation learning [Schölkopf et al., 2021, Schölkopf and von Kügelgen, 2022] is an emerging field that exploits ideas from the theory of latent variable modeling [Arora et al., 2013, Kivva et al., 2021, Hyvärinen et al., 2023] along with causality [Spirtes et al., 2000, Pearl, 2009, Rajendran et al., 2021, Squires and Uhler, 2022] to build generative models for various domains. The main goal is to build the true generative models that led to the creation of a dataset. This field has made exciting advances in recent years [Khemakhem et al., 2020, Falck et al., 2021, Zimmermann et al., 2021, Kivva et al., 2022, Lachapelle et al., 2022, Rajendran et al., 2023, Varici et al., 2023, Rajendran et al., 2024a, Jiang and Aragam, 2023, Buchholz et al., 2023, Hyvärinen et al., 2023]. However, to study large foundation models, a purely statistical notion of building the true model may not reflect the entire story as one should also take into account the implicit bias of optimization as well (Section 2.3.2). Moreover, identifying the entire underlying latent distributions may also not be necessary for certain practical purposes and it might be sufficient to represent only certain structure information in the geometry of representations, such as linearly encoded representations. Another example of this has been recently explored by [Jiang et al., 2023] namely independence preserving embedding which studies how independence structure can be stored in representations.

2.7 Conclusion

In this chapter, we presented a simple latent variable model and showed that using a standard LLM pipeline to learn the distribution results in concepts being linearly represented. We observed that this linear structure is promoted in (at least) two ways—(i) matching log-odds, similar to prior works on word embeddings, and (ii) implicit bias of gradient descent. Additionally, experimental results show that—as predicted—the linear structure emerges from the simple latent variable model. We also saw that, as predicted in the simple model, LLaMA-2 representations exhibit alignment between embedding and unembedding representations.

CHAPTER 3

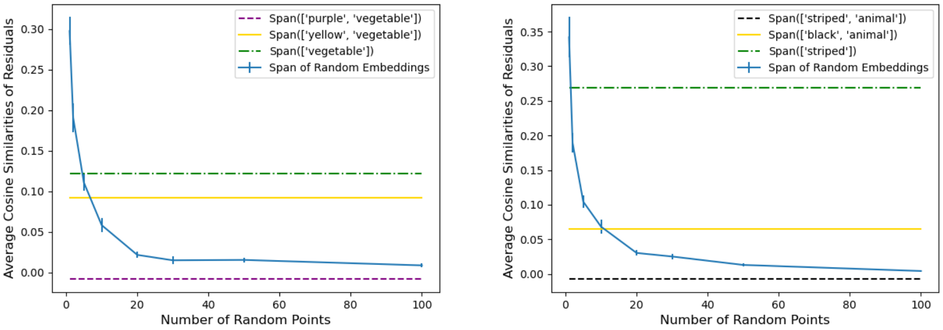
ALGEBRAIC STRUCTURE AND PARTIAL ORTHOGONALITY

3.1 Introduction

This chapter concerns the question of how semantic meaning is encoded in neural embeddings, such as those produced by [Radford et al., 2021]. There is strong empirical evidence that these embeddings—vectors of real numbers—capture the semantic meaning of the underlying text. For example, classical results show that word embeddings can be used for analogical reasoning [e.g., Mikolov et al., 2013a, Pennington et al., 2014], and such embeddings are the backbone of modern generative AI systems [e.g., Ramesh et al., 2022, Bubeck et al., 2023, Saharia et al., 2022, Devlin et al., 2018]. The high-level question we're interested in is: *How is the* ***semantic*** *structure of text encoded in the* ***algebraic*** *structure of embeddings?* In this chapter, we provide evidence that the concept of *partial orthogonality* plays a key role.

The first step is to identify the semantic structure of interest. Intuitively, words or phrases possess a notion of semantic independence, which does not have to be statistical in nature. For example, the word “eggplant” seems more similar to “tomato” than to “ennui”. Yet, if we were to “condition" on the common property of “vegetable”, then “eggplant” and “tomato” should be “independent". And, if we condition on both “vegetable” and “purple”, then “eggplant" may be “independent” of all other words. However, it is difficult to formalize what is meant by “independent" and “condition on" in these informal statements. Accordingly, it is hard to establish a formal definition of semantic independence, and thus it is challenging to explore how this structure might be encoded algebraically!

The key observation in this chapter is to recall that most reasonable concepts of “independence” adhere to a common set of axioms similar to those defining probabilistic conditional independence. Formally, this abstract idea is captured by the axioms of the so-called *inde-*



(a) 'eggplant'

(b) 'zebra'

Figure 3.1: For target embedding of "eggplant", the set of embeddings that include "purple" and "vegetable" forms the subspace such that after projection, the residual of 'eggplant" has the lowest cosine similarity with residuals of other test embeddings. This matches our intuition for the meaning of "eggplant". Similarly, for target embedding of "zebra", the set of embeddings that include "striped" and "animal" forms the most suitable subspace.

pendence models [Lauritzen, 1996]. Thus, if semantic independence is encoded algebraically, it should be encoded as an algebraic structure that respects these axioms. In this chapter, we use a natural candidate independence model in vector spaces known as *partial orthogonality* [Lauritzen, 1996, Amini et al., 2022]. Here, for two vectors v_a and v_b and a conditioning set of vectors v_C, partial orthogonality takes v_a independent v_b given v_C if the residuals of v_a and v_b are orthogonal after projecting onto the span of v_C. *We discover that this particular tool is indeed valuable for understanding CLIP embeddings.* For instance, Figure 3.1 shows that after projecting onto the linear subspace spanned by CLIP embeddings of "purple" and "vegetable", the residual of embedding "eggplant" has on average low cosine similarity with the residuals of random test embeddings, which also matches our intuitive understanding of the word.

Since partial orthogonality is an independence model, we can go one step further to define *Markov boundaries* for embeddings as well. Drawing inspiration from graphical models, it is reasonable to expect that the Markov boundary of any target embedding should constitute a minimal collection of embeddings that encompasses valuable information regarding the target. Unlike classical applications of partial orthogonality in regression and Gaussian models, however, the geometry of embeddings presents several subtle technical challenges to directly adopting the usual notion of Markov boundary. First, the *intersection axiom* never holds for practical embeddings, which makes the standard Markov boundary non-unique. More importantly, practical embeddings could potentially incorporate distortion, noise and undergo phenomena resembling superposition [Elhage et al., 2022a]. Therefore, in this chapter, we introduce *generalized Markov boundaries* for studying the structure of text embeddings.

Contributions Specifically, we make the following contributions:

1. We adapt ideas from graphical independence models to specify the structure that should be satisfied by semantic independence. We discover that partial orthogonality in the embedding space offers a natural way of encoding semantic independence structure (Section 3.2).

2. We study the semantic structure of partial orthogonality via Markov boundaries. Due to the unique characteristics of embeddings and noise in learning, exact orthogonality is unlikely to hold. So, we give a distributional relaxation of the Markov boundary and use this to provide a practical algorithm for finding generalized Markov boundaries and measuring the semantic independence induced by generalized Markov boundaries (Section 3.3.2).

3. We introduce the concept of *independence preserving embeddings*, which studies how embeddings can be used to maintain the independence structure of distributions. This holds its own intrigue for further research (Section 3.4).

4. Finally, we design and conduct experimental evaluations on CLIP text embeddings, finding that the partial orthogonality structure and generalized Markov boundary encode semantic structure (Section 4.5).

Throughout, we use CLIP text embeddings as a running example, though the method and theory presented can be applied more broadly.

3.2 Independence model and Markov boundary

Let E be a finite set of embeddings with $|\,\mathrm{E}\,| = n$ and each embedding is of size d. Every embedding is a vector representation of a word. In other words, there exists a function f that maps words to n vectors in $\mathbb{R}^d$. As explained above, we might expect embeddings to encode "independence structures" between words. These independence structures are difficult to define formally, though the structure is similar to that of probabilistic conditional independence. We will use independence models as an abstract formalization of this structure.

3.2.1 Independence model

Throughout this chapter, we use many standard definitions and facts about graphical models and more generally, abstract independence models. A detailed overview of this material can be found, for instance, in [Lauritzen, 1996, Studený, 2005].

Suppose V is a finite set. In the case of embeddings, V would be a set of vectors. An *independence model* $\perp\!\!\!\perp_\sigma$ is a ternary relation on V. Let A, B, C, D be disjoint subsets of V. Then a *semi-graphoid* is an independence model that satisfies the following axioms:

(A1) (Symmetry) If $A \perp\!\!\!\perp_\sigma B|C$, then $B \perp\!\!\!\perp\, _\sigma A|C$;

(A2) (Decomposition) If $A \perp\!\!\!\perp_\sigma (B \cup D)|C$, then $A \perp\!\!\!\perp_\sigma B|C$ and $A \perp\!\!\!\perp_\sigma D|C$;

(A3) (Weak Union) If $A \perp\!\!\!\perp_\sigma (B \cup D)|C$, then $A \perp\!\!\!\perp_\sigma B|(C \cup D)$;

(A4) (Contraction) If $A \perp\!\!\!\perp_\sigma B|C$ and $A \perp\!\!\!\perp_\sigma D|(B \cup C)$, then $A \perp\!\!\!\perp_\sigma (B \cup D)|C$.

The independence model is a *graphoid* if it also satisfies

(A5) (Intersection) If $A \perp\!\!\!\perp_\sigma B|(C \cup D)$ and $A \perp\!\!\!\perp_\sigma C|(B \cup D)$, then $A \perp\!\!\!\perp_\sigma (B \cup C)|D$.

And, the graphoid is called a *compositional graphoid* if it also satisfies

(A6) (Composition) If $A \perp\!\!\!\perp_\sigma B|C$ and $A \perp\!\!\!\perp_\sigma D|C$, then $A \perp\!\!\!\perp_\sigma (B \cup D)|C$.

We also use $\mathcal{I}_\sigma(V)$ to be the set of conditional independent tuples under the independence model $\perp\!\!\!\perp_\sigma$. In other words, if $(A, B, C) \in \mathcal{I}_\sigma(V)$, then $A \perp\!\!\!\perp_\sigma B|C$ where A, B, C are disjoint subsets of V.

Probabilistic Conditional Independence ($\perp\!\!\!\perp_\mathrm{P}$) Given a finite set of random variables V, probabilistic conditional independence over V defines an independence model that satisfies (A1)-(A4) which means that probabilistic independence models are semi-graphoids. In general, however, they are not compositional graphoids. If the distribution has strictly positive density w.r.t. a product measure, then the intersection axiom is true. In this case, probabilistic independence models are graphoids. Still, in general, the composition axiom is not satisfied because pairwise independence does not imply joint independence. One notable exception is when the distribution is regular multivariate Gaussian; then the probabilistic independence model is a compositional graphoid.

Undirected Graph Separations ($\perp\!\!\!\perp_\mathrm{G}$) For a finite undirected graph $\mathcal{G} = (V, E)$. One can easily show that ordinary graph separation in undirected graphs is a compositional graphoid. The relations between probabilistic conditional independences and graph separations are well-studied in the graphical modeling literature [Koller and Friedman, 2009, Lauritzen, 1996]. We recall a few important definitions here for completeness. Consider a natural bijection between graphical nodes and random variables. Then if $\mathcal{I}_G(V) \subseteq \mathcal{I}_P(V)$, we say the distribution $\mathcal{P}$ over V satisfies the *Markov property* with respect to $\mathcal{G}$ and $\mathcal{G}$ is called an *I-map* of $\mathcal{P}$. An I-map $\mathcal{G}$ for $\mathcal{P}$ is minimal if no subgraph of $\mathcal{G}$ is also an I-map of $\mathcal{P}$. It is not difficult to show that there exists a minimal I-map $\mathcal{G}$ for any distribution $\mathcal{P}$.

Remark 10. Not every compositional graphoid can be represented by an undirected graph. Sadeghi [2017] provides sufficient and necessary conditions for this.

Partial Orthogonality ($\perp\!\!\!\perp_{\mathrm{O}}$) Let V be a finite collection of vectors in $\mathbb{R}^d$. If $a \in V, b \in V$ and $C \subseteq V$, then we say that a and b are *partially orthogonal given* C if

$$a \perp\!\!\!\perp_{\mathrm{O}} b|C \iff \left\langle \mathrm{proj}_C^{\perp}[a],\ \mathrm{proj}_C^{\perp}[b] \right\rangle = 0,$$

where $\mathrm{proj}_C^{\perp}[a] = a - \mathrm{proj}_C[a]$ is the residual of a after projection onto the span of C. It is not hard to verify that $\perp\!\!\!\perp_{\mathrm{O}}$ is a semi-graphoid that also satisfies the composition axiom (A6). When V is a set of linearly independent vectors, then $\perp\!\!\!\perp_{\mathrm{O}}$ satisfies (A5) and thus is a compositional graphoid. Partial orthogonality has been studied under different names in the statistics literature for many decades. For example, if we replace Euclidean space with the L^2 space of random variables, partial orthogonality is equivalent to the well-known concept of *partial correlation* or *second-order independence* (Example 2.26 in Lauritzen [2020]). The concept of geometric orthogonality (Example 2.27 in Lauritzen [2020]) is closely related but does not always satisfy the intersection axiom. More recently, the concept of partial orthogonality in abstract Hilbert spaces was defined and studied extensively in Amini et al. [2022]. Finally, when V is a linearly independent collection of vectors, partial orthogonality yields a stronger independence model known as a *Gaussoid*, which is well-studied [e.g. Lněnička and Matúš, 2007, Boege and Kahle, 2019, and the references therein]. It is worth emphasizing that in the present setting of text embedding, we typically have $d \ll n$, and hence V cannot be linearly independent.

3.2.2 Markov boundaries

Suppose $\perp\!\!\!\perp_\sigma$ is an independence model over a finite set V. Let v_i be an element in V, then the Markov blanket $\mathcal{M}$ of v_i is any subset of $V \setminus \{v_i\}$ such that

$$v_i \perp\!\!\!\perp_\sigma V \setminus (\{v_i\} \cup \mathcal{M}) | \, \mathcal{M}$$

A *Markov boundary* is a minimal Markov blanket.

A Markov boundary, by definition, always exists and can be an empty set. However, it might not be unique. It is well-known that *the intersection property is a sufficient condition to guarantee Markov boundaries are unique.* Thus, the Markov boundary is unique in any graphoid. The proof is presented here for completeness.

Theorem 11. *If $\perp\!\!\!\perp_\sigma$ is a graphoid over V, then the Markov boundary is unique for any element in V.*

Proof. Let $v_i \in V$. Suppose v_i has two distinct Markov boundaries $\mathcal{M}_1$, $\mathcal{M}_2$. Then they must be non-empty and $v_i \not\perp\!\!\!\perp_\sigma \mathcal{M}_1$, $v_i \not\perp\!\!\!\perp_\sigma \mathcal{M}_2$, $v_i \perp\!\!\!\perp_\sigma \mathcal{M}_2 \,|\, \mathcal{M}_1$, $v_i \perp\!\!\!\perp_\sigma \mathcal{M}_1 \,|\, \mathcal{M}_2$. By the intersection axiom, $v_i \perp\!\!\!\perp_\sigma \mathcal{M}_1 \cup \mathcal{M}_1$. Then by the decomposition axiom, $v_i \perp\!\!\!\perp_\sigma \mathcal{M}_1$ and $v_i \perp\!\!\!\perp_\sigma \mathcal{M}_2$ which is a contradiction. □

Remark 12. For any semi-graphoid, the intersection property is not a necessary condition for the uniqueness of Markov boundaries. See Remark 1 in Wang and Wang [2020].

The connection between orthogonal projection and graphoid axioms is well-known [Lauritzen, 1996, Dawid, 2001, Whittaker, 2009]. But graphoid axioms find their primary applications in graphical models [Lauritzen, 1996]. In particular, there are many existing papers on Markov boundary discovery for graphical models [Tsamardinos et al., 2003b, Aliferis et al., 2010, Strobl and Visweswaran, 2016, Gao and Aragam, 2021, Tsamardinos et al., 2003a, Pena et al., 2007]. They typically assume faithfulness or the distributions are strictly positive, which are sufficient conditions for the intersection property and thus ensure unique

Markov boundaries. As an important axiom for graphoids, the intersection property has also been thoroughly investigated [San Martin et al., 2005, Peters, 2015, Fink, 2011]. But the intersection property rarely holds for embeddings (See Section 3.3), which means there could be multiple Markov boundaries. Statnikov et al. [2013], Wang and Wang [2020] study this case for graphical models and causal inference.

3.3 Markov boundary of embeddings

As indicated in Section 3.2, partial orthogonality ($\perp\!\!\!\perp_{\mathrm{O}}$) can be used as an independence model over vectors in Euclidean space and is a compositional semi-graphoid. Thus, one can use partial orthogonality to study embeddings, which are real vectors. When $n \leq d$ and the vectors in E are linearly independent, every vector in E has a unique Markov boundary by Theorem 11.

Unfortunately, when $d < n$, which happens in practice with embeddings as there are usually more objects to embed than the embedding dimension, there is a possibility of having multiple Markov boundaries. In fact, the main challenge with Markov boundary discovery for embeddings is that *the intersection property generally does not hold*, as opposed to graphical models where this property is commonly assumed [Tsamardinos et al., 2003b, Aliferis et al., 2010, Strobl and Visweswaran, 2016].

While the Markov boundary might not be unique, the following theorem says that all Markov boundaries of the target vector capture the same *"information"* about that vector.

Theorem 13. *Let partial orthogonality* $\perp\!\!\!\perp_{\mathrm{O}}$ *be the independence model over a finite set of embedding vectors* E. *Suppose* $\mathcal{M}_1, \mathcal{M}_2 \subseteq \mathrm{E}$ *are two distinct Markov boundaries of* $v_i \in \mathrm{E}$, *then,*

$$proj_{\mathcal{M}_1}[v_i] = proj_{\mathcal{M}_2}[v_i]$$

When $d \ll n$, then it is likely that the target embedding v_i lies in the linear span of other embeddings (i.e, $v_i \in \text{span}(\mathrm{E} \setminus \{v_i\})$), Corollary 14 below shows that, in this case, the span of any Markov boundary is precisely the subspace that contains v_i:

Corollary 14. *Let parital orthogonality $\perp\!\!\!\perp_{\mathrm{O}}$ be the independence model over a finite set of embedding vectors* E*. Suppose $\mathcal{M}_1 \subseteq \mathrm{E}$ is a Markov boundary of $v_i \in \mathrm{E}$ and $v_i \in \text{span}(\mathrm{E} \setminus \{v_i\})$, then,*

$$proj_{\mathcal{M}_1}[v_i] = v_i.$$

In other words, to find a Markov boundary of v_i, we need to find some vectors such that their linear combination is exactly v_i. This seems very strict but is necessary because the formal definition of the Markov boundary requires residual orthogonalities between v_i and every other vector. In the sequel, we show how to relax the definition of the Markov boundary.

3.3.1 From elementwise orthogonality to distributional orthogonality

Corollary 14 suggests that the span of the Markov boundary for any target vector should contain that target vector. This is a consequence of the elementwise orthogonality constraint because the definition of the Markov boundary requires the residual of a target vector to be orthogonal to the residual of any test vector. The implicit assumption here is that embeddings are distortion-free and every non-zero correlation is meaningful. However, due to the inherent limitation of the embedding dimension—which often restricts the available space for storing all the orthogonal vectors—and noises introduced from training, embeddings are likely prone to distortion when compressed into a relatively small Euclidean space. In fact, we empirically show in Section 3.5.2 that inner products in embedding space do not necessarily respect semantic meanings faithfully. Therefore, the notion of elementwise orthogonality loses practical significance.

Instead of enforcing elementwise orthogonality, we relax the definition of the Markov boundary of embeddings such that intuitively, after projection, the residual of the target vector and the residuals of test vectors should be orthogonal in a distributional sense where the distribution is the empirical distribution over test vectors. To capture distributional orthogonalities, this chapter focuses on the average of cosine similarities.

In particular, we have the following definition of *generalized Markov boundary* for partial orthogonality.

Definition 15 (Generalized Markov Boundary for Partial Orthogonality)**.** Given a finite set E of embedding vectors. Let v be an element in E, then a *generalized Markov boundary* $\mathcal{M}$ of v is a minimal subset of $\mathrm{E}\setminus\{v\}$ such that

$$\mathrm{S}_{\mathcal{M}}(v, \mathrm{E}) = \frac{1}{|\,\mathrm{E}^v_{\mathcal{M}}\,|} \sum_{u \in \mathrm{E}^v_{\mathcal{M}}} \mathrm{S^c}_{\mathcal{M}}(v, u) = 0$$

where $\mathrm{S^c}_{\mathcal{M}}(v, u)$ is the cosine similarity of u and v after projection and $\mathrm{E}^v_{\mathcal{M}} = \mathrm{E}\setminus(\{v\}\cup\mathcal{M})$. Specifically, $\mathrm{S^c}_{\mathcal{M}}(v, u) = \frac{\langle \mathrm{proj}^{\perp}_{\mathcal{M}}[v], \mathrm{proj}^{\perp}_{\mathcal{M}}[u] \rangle}{||\mathrm{proj}^{\perp}_{\mathcal{M}}[v]|| \cdot ||\mathrm{proj}^{\perp}_{\mathcal{M}}[u]||}$.

Intuitively, this suggests that, on average, there is no particular direction of residuals that have nontrivial correlations with the residual of the target embedding.

Remark 16. It is evident that the conventional definition of Markov boundary implies Definition 15 (Lemma 35 in Appendix B.1).

3.3.2 Finding generalized Markov boundary

With a formal definition of the generalized Markov boundary established, our objective is now to identify this boundary. One can always use brute force by enumerating all possible subsets of E, but the algorithm would be infeasible when $|\,\mathrm{E}\,|$ is large.

Suppose $v \in \mathrm{E}$ is a target vector and $\mathcal{M}$ is its generalized Markov boundary, then we can write $v = v_{\perp} + v_{\|}$ where $v_{\perp} = \mathrm{proj}^{\perp}_{\mathcal{M}}[v]$ and $v_{\|} = \mathrm{proj}_{\mathcal{M}}[v]$. Intuitively, Definition 15 suggests that the residual of test vectors can appear in any direction relative to $v_{\perp}$. Therefore, if one samples random test vectors $\{u_i\}$, their span is likely to be close to $v_{\perp}$. In other words,

Algorithm 1 Approximate Algorithm to Find Generalized Markov Boundary

```
Input: $v$, $\mathrm{E}$
/* $\mathrm{E}$ is the set of all embeddings and $v \in \mathrm{E}$ is the target embedding */
Input: $n_r$, $d_r$, $K$
/* $n_r$ is the number of sampled random subspaces, $d_r$ is the number of sampled
   vectors for each random subspace and $K$ is the number of candidate
   vectors to construct generalized Markov boundary */
Output: $\mathcal{M} \subseteq \mathrm{E}$
/* $\mathcal{M}$ is the estimated generalized Markov boundary for $v$ */
for $i \leftarrow 1$ to $n_r$ do
    randomly sample a set of $d_r$ vectors $\mathcal{M}_i = \{v_k^i\}_{k=1}^{d_r} \subseteq (\mathrm{E} \setminus \{v\})$
     caculate ${\mathrm{S}^{\mathrm{c}}}_{\mathcal{M}_i}(v, u)$ for all $u \in (\mathrm{E} \setminus \{v\})$
end
Find the top $K$ vectors $\{u_i\}_{i=1}^{K}$ with the highest $\sum_i {\mathrm{S}^{\mathrm{c}}}_{\mathcal{M}_i}(v, u)$ .
Find the subset $\mathcal{M}$ of $\{u_i\}_{i=1}^{K}$ that has the lowest $\mathrm{S}_{\mathcal{M}}(v, \mathrm{E})$.
Result: $\mathcal{M}$
```

the residual of v after projection onto span($\{u_i\}$) should contain more information about the generalized Markov boundary direction $v_{\|}$.

This motivates the approximate method Algorithm 1. For any target embedding v, one first sample subspaces spanned by randomly selected embeddings. Embeddings that, on average have high cosine similarities with the target embedding after projecting onto orthogonal complements of previously sampled random subspaces, are considered to be candidates for the generalized Markov boundary. The final selection of generalized Markov boundary searches over these top K candidates.

Empirically, for text embedding models like CLIP, random projections prove to be advantageous in revealing semantically related concepts. In Section 3.5.2, we provide several examples where, for a given target embedding, the embeddings that exhibit high correlation after random projections are more semantically meaningful compared to embeddings with merely high cosine similarity with the target embedding before projections.

3.4 Independence preserving embeddings (IPE)

In the previous sections, we discussed the Markov boundary of embeddings under the partial orthogonality independence model. In Section 4.5, we will test its effectiveness at capturing the "semantic independence structure" through experiments conducted on CLIP text embeddings. The belief is that the linear algebraic structure possesses the capacity to uphold the independence structure of semantic meanings.

A natural question to ask is: *Is it always possible to use vector space embeddings to preserve independence structures of interest?* In this section, we study the case for random variables. Consider an embedding function f that maps a random variable X to $f(X) \in \mathbb{R}^d$. Ideally, it is desirable for the partial orthogonalities of embeddings to mirror the conditional independences present in the joint distribution of X. We call such representations *independence preserving embeddings (IPE)* (Definition 17). In this section, we delve into the theoretical feasibility of these embeddings by initially demonstrating the construction of IPE and then showing how one can use random projection to reduce the dimension of IPE. We believe that studying IPE lays the theoretical foundation to understand embedding models in general.

Definition 17 (Independence Preserving Embedding Map)**.** Let V be a finite set of random variables with distribution P. A function $f : V \to \mathbb{R}^d$ is called an *independence preserving embedding map* (IPE map) if

$$\mathcal{I}_O(f(V)) \subseteq \mathcal{I}_P(V).$$

An IPE map is called a *faithful* IPE map if

$$\mathcal{I}_O(f(V)) = \mathcal{I}_P(V).$$

3.4.1 Existence and universality of IPE maps

We first show that for *any distribution* P over random variables V, we can construct an IPE map.

For any distribution P over V, there exists a minimal I-map $\mathcal{G} = (V, E)$ such that $\mathcal{I}_G(V) \subseteq \mathcal{I}_P(V)$ (See Section 3.2). We will use $\mathcal{G}_P$ to be a minimal I-map of P and $\operatorname{adj}(\mathcal{G}_P)$ to be the adjacency matrix of $\mathcal{G}_P$. We further define $\operatorname{adj}_\varepsilon(\mathcal{G}_P)$ to be an *adjusted adjacency matrix* with $\varepsilon \in \mathbb{R}$ where

$$\operatorname{adj}_\varepsilon(\mathcal{G}_P) = \mathbb{1} + \varepsilon \operatorname{adj}(\mathcal{G}_P)$$

and $\mathbb{1}$ is the identity matrix.

Ideally, this matrix is invertible, however, it turns out that not every ε produces an invertible $\operatorname{adj}_\varepsilon(\mathcal{G}_P)$. We therefore define the following *perfect perturbation factor*. For any matrix $A \in \mathbb{R}^{n \times n}$, define $A_{\mathcal{I},\mathcal{J}}$ to be the submatrix of A with row and column indices from $\mathcal{I}$ and $\mathcal{J}$, respectively. If $\mathcal{I} = \mathcal{J}$, the submatrix is called a *principal submatrix* and we denote it simply as $A_{\mathcal{I}}$.

Definition 18 (Perfect Perturbation Factor)**.** For a given graph $\mathcal{G} = (V, E)$ where $n = |V|$, ε is called a *perfect perturbation factor* if (1) $\operatorname{adj}_\varepsilon(\mathcal{G}_P)$ is invertible and (2) for any $\mathcal{I} \subseteq [n]$, $(\operatorname{adj}_\varepsilon(\mathcal{G}_P)_{\mathcal{I}})^{-1}_{ij} = 0$ if and only if $v_{\mathcal{I}_i} \perp\!\!\!\perp_{\mathrm{G}} v_{\mathcal{I}_j} | \{v_k : k \notin \mathcal{I}\}$ where $\mathcal{I}_i$ is the ith element of $\mathcal{I}$.

Theorem 19. *Let V be a finite set of random variables with distribution P. $\mathcal{G}_P$ is a minimal I-map of P. Let A be equal to $\operatorname{adj}_\varepsilon(\mathcal{G}_P)^{-1}$ with eigen decomposition $A = U\Sigma U^T$. If ε is a perfect perturbation factor, then the function f with*

$$f(v_i) = U_i \Sigma^{1/2}$$

is an IPE map of P where v_i is a random variable in V and U_i is the i-th row of U.

Furthermore, if P is faithful to $\mathcal{G}_P$, then f is a faithful IPE map for $\mathcal{P}$.

Remark 20. One can always normalize these embeddings to have unit norms without changing the partial orthogonality structures.

Finding a perfect perturbation factor might seem daunting, but the following lemma, which is a direct consequence of Theorem 1 in Lněnička and Matúš [2007], shows that almost every ε is a perfect perturbation factor.

Lemma 21. *For any simple graph G, ε is perfect for all but finitely many $\varepsilon \in \mathbb{R}$.*

3.4.2 Dimension reduction of IPE

Theorem 19 shows how to learn a perfect IPE but it requires the dimension of embeddings to be the same as the number of variables in V. In the worst case, this is inevitable for a faithful IPE map: If the random variables in V are mutually independent, then we need at least $|V|$ dimensions in the embedding space to contain V orthogonal vectors.

But this is not practical. Suppose we want to embed millions of random variables (e.g. tokens) in a vector space, having the dimension of each embedding be in the magnitude of millions is less than ideal. Therefore, one needs to do dimension reduction.

In this section, we show that by using random projection, the partial orthogonalities induced by Markov boundaries are preserved approximately. Intuitively, this is guaranteed by the Johnson-Lindenstrauss lemma Vempala [2005].

Theorem 22. *Let U be a set of vectors in $\mathbb{R}^n$ where $n = |U|$ and every vector is a unit vector. Let Σ be a matrix in $\mathbb{R}^{n\times n}$ where $\Sigma_{ij} = \langle u_i, u_j\rangle$. Assume $\lambda_1 = \lambda_{\min}(\Sigma) > 0$. Then there exists a mapping $g : \mathbb{R}^n \to \mathbb{R}^k$ where $k = \lceil 20\log(2n)/(\varepsilon')^2\rceil$ with $\varepsilon' = \min\{1/2, \varepsilon/C, \lambda_1/2r^2\}$ and $\varepsilon \in (0, 1)$ such that for any $u_i \in U$ with its unique Markov boundary $M_i \subseteq U$ and any*

$u_j \in U \setminus (\{u_i\} \cup M_i)$, we have

$$\left| \left\langle \mathrm{proj}^{\perp}_{g(M_i)}[g(u_i)], \mathrm{proj}^{\perp}_{g(M_i)}[g(u_j)] \right\rangle \right| \leq \varepsilon$$

where $r_i = |M_i|$, $r = \max_i |M_i|$ and $C = (r+1)^3(\frac{2\lambda_{\max}(\Sigma)+2(r+1)^2}{\lambda_{\min}(\Sigma)})^r$.

Theorem 22 shows that as long as the partial orthogonality structure of embeddings is sparse in the sense that the size of the Markov boundary for each embedding is small. Then one can reduce the dimension of the embedding and the residuals of target and test vectors after projection onto the Markov boundary are *almost orthogonal*.

Remark 23. The assumption in Theorem 22 is satisfied by the construction of IPE in Section 3.4.1.

3.5 Experiments

One of the central hypotheses of the chapter is that the partial orthogonality of embeddings, and its byproduct generalized Markov boundary, carry semantic information. To verify this claim, we provide both *quantitative* and *qualitative* experiments. Throughout this section, we consider the set of normalized embeddings E that represent the 49815 words in the Brown corpus Francis and Kucera [1979]. For each target embedding of a word, under any experiment setting, we automatically filter words, whose embeddings have 0.9 or above cosine similarities with the target embedding, or words, whose Wu-Palmer similarity measure with the target word is almost 1. The purpose of this filtering step is to prevent the inclusion of synonyms.

3.5.1 Semantic structure of partial orthogonality

To examine the rule of partial orthogonality, nine categories are chosen, each with 10 words in it (See Table B.1 in Appendix B.2). Specifically, each word within a given category is a hyponym for that category in WordNet Miller [1995]. We assess how much, on average, the cosine similarities between words within each category decrease when conditioned on these different nine categories. By conditioning, we use the clip embedding of the category word of interest and project out the subspace of that clip embedding. The results are shown in Figure 3.2. We normalize reduction values by sampling 10,000 embeddings and calculating the mean and standard deviation of cosine reductions between these embeddings. It is apparent that on average, cosine similarities of intra-category words decrease more than inter-category words. One interesting finding is that when conditioned on the category word "food", the average similarities between word pairs in "beverage" also drop considerably. We suspect this is because one synset of "food" is also a hypernom of "beverage". Although words in the "food" category are chosen to mean solid food, it could also mean nutrient which also encompasses the meaning of "beverage".

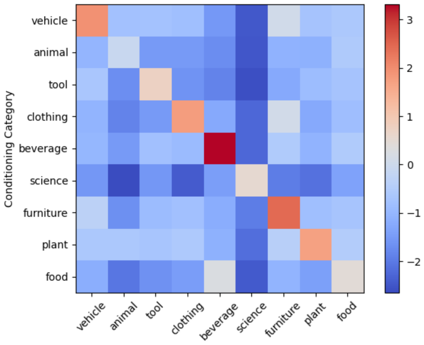


Figure 3.2: Experiments show conditioning on the category word, cosine similarities of intra-category words decrease more than inter-category words. Each row shows the (normalized) average cosine similarities reduction between words within each category when conditioned on the category word of that row.

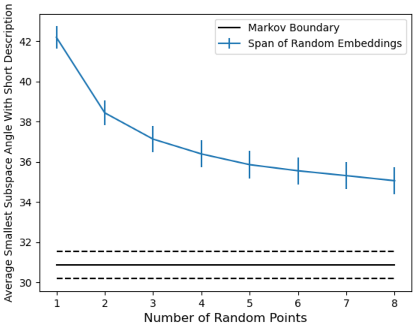


Figure 3.3: Experiments show that learned generalized Markov boundaries have on average smaller principal angles with the description embedding compared to subspaces spanned by randomly selected embeddings. The standard errors are over 50 examples.

3.5.2 Sampling random subspaces

The first step of Algorithm 1 is to find embeddings that have high similarities with the target embeddings even after projecting onto orthogonal complements of subspaces spanned by randomly selected embeddings. It turns out that this step can reveal semantic meanings. In this section, we design experiments to show both quantitatively and qualitatively that embeddings of words that remain highly correlated with the target embedding after projection are semantically closer to the target word. In various experimental configurations, we employ 10 sets of 50 randomly chosen embeddings to form random projection subspaces for each target embedding. Qualitatively, Table B.2 in Appendix B.2 gives a few examples showing that the words that on average remain highly correlated with the target word tend to possess greater semantic significance. Quantitatively, we calculate the average Wu-Palmer similarities between target words and the top 10 correlated words before and after random projections. We conduct experiments on 1000 random words as well as 300 common nouns provided by ChatGPT. The results are shown in Table B.2 verify our claims. This set of experiments also indirectly shows that the embeddings are noisy and that generalized Markov boundaries are indeed needed.

Table 3.1: Experiments show that the top 10 words that have, on average, high correlations with target words after projecting onto the orthogonal complements of randomly selected linear subspaces have higher Wu-Palmer similarities with the target words than the top 10 highly correlated words without projections. This table contains average Wu-Palmer similarities with standard errors.

Target	Before Projection	After Projection
Random Words	0.223 ± 0.006	0.245 ± 0.007
Common Nouns	0.343 ± 0.008	0.422 ± 0.008

3.5.3 Generalized Markov boundaries

We first demonstrate that Algorithm 1 can find generalized Markov boundaries. The experiments are run over 1000 randomly selected words. In particular, Table 3.2 shows that with a relatively small candidate set, the algorithm can already approximate generalized Markov boundaries well, suggesting that the size of generalized Markov boundaries for CLIP text embeddings should be small.

Semantic Meanings of Markov Boundaries The estimated generalized Markov boundaries returned by Algorithm 1 is a set of embeddings. It is reasonable to anticipate that the linear spans of these embeddings hold semantic meanings. To evaluate this hypothesis, we propose to calculate the smallest principal angles [Knyazev and Argentati, 2002] between the span of generalized Markov boundaries and the span of selected embeddings that are meaningful to the target word.

We again conducted both quantitative and qualitative experiments. Qualitatively, Figure B.1 in Appendix B.2 give a few examples comparing target words' generalized Markov boundaries with the span of selected embeddings. For instance, the generalized Markov boundary of 'car' is more aligned with the subspace spanned by embeddings of 'road' and 'vehicle' than the span of 'sea' and 'boat' and randomly selected subspaces. This suggests that the estimated generalized Markov boundaries hold semantic significance. To verify this quantitatively, we ask ChatGPT to provide a list of common nouns with short descriptions (selected examples are provided in Table B.3). We then use CLIP text embedding to convert the description sentence into one vector and compare the smallest angle between the description vector with generalized Markov boundaries and random linear spans. Figure 3.3 shows that the generalized Markov boundaries are more semantically meaningful than random subspaces.

Table 3.2: With relatively small number of K, the average $\mathrm{S}_{\mathcal{M}}(v, \mathrm{E})$ is small. The standard errors are over 1000 experiments.

K	1	3	5	8	10
Average $\mathrm{S}_{\mathcal{M}}(v, \mathrm{E})$	0.345±0.03	0.128±0.03	0.054±0.002	0.015±0.002	0.008±0.001

3.6 Related work

There are many papers [e.g., Arora et al., 2016, Gittens et al., 2017b, Allen and Hospedales, 2019, Ethayarajh et al., 2019, Trager et al., 2023, Perera et al., 2023, Leemann et al., 2023, Merullo et al., 2023, Wang et al., 2023d] connecting semantic meanings and algebraic structures of popular embeddings like CLIP [Radford et al., 2021], Glove [Pennington et al., 2014] and word2vec Mikolov et al. [2013a]. Simple arithmetic on these embeddings reveals that they carry semantic meanings. The most popular arithmetic operation is called linear analogy [Ethayarajh et al., 2019]. There are several papers trying to understand the reasoning behind this phenomenon. Arora et al. [2016] explains this by proposing the latent variable model but it requires the word vectors to be uniformly distributed in the embedding space which generally is not true in practice [Mimno and Thompson, 2017b]. Alternatively, Gittens et al. [2017b], Allen and Hospedales [2019] adopts the paraphrase model that also does not fit practice. Ethayarajh et al. [2019], on the other hand, studies the geometry of embeddings that decomposes the shifted pointwise mutual information (PMI) matrix. Trager et al. [2023], Perera et al. [2023] decomposes embeddings into combinations of a smaller set of vectors that are more interpretable. On the other hand, similar to using vector orthogonality to represent (conditional) independence, kernel mean embeddings [Muandet et al., 2017] are Hilbert space embeddings of distributions that can also be used to represent conditional independences [Song et al., 2009, 2013]. It is a popular method for machine learning, and causal inference [Gretton et al., 2005, Mooij et al., 2009, Greenfeld and Shalit, 2020]. But unlike independence preserving embeddings, kernel mean embeddings use the kernel and do not explicitly construct finite-dimensional vector representations.

3.7 Conclusion

This chapter studies the role of partial orthogonality in analyzing embeddings. Specifically, we extend the idea of Markov boundaries to embedding space. Unlike Markov boundaries in graphical models, the boundaries for embeddings are not guaranteed to be unique. We propose alternative relaxed definitions of Markov boundaries for practical use. Empirically, these tools prove to be useful in finding the semantic meanings of embeddings. We also introduce the concept of independence preserving embeddings where embeddings use partial orthogonalities to preserve the conditional independence structures of random variables. This opens the door for substantial future work. In particular, one promising theoretical direction is to study if CLIP text embeddings preserve the structures in the training distributions.

CHAPTER 4

COMPOSITE STRUCTURE AND ASSOCIATIVE MEMORY

4.1 Introduction

What is the first thing that would come to mind if you were asked *not* to think of an elephant? Chances are, you would be thinking about elephants. What if we ask the same thing to Large Language Models (LLMs)? Obviously, one would expect the outputs of LLMs to be heavily influenced by tokens in the context [Brown et al., 2020]. Could such influence potentially prime LLMs into changing outputs in a nontrivial way? To gain a deeper understanding, we focus on one specific task called fact retrieval [Meng et al., 2022, 2023] where expected output answers are given. LLMs, which are trained on vast amounts of data, are known to have the capability to store and recall facts [Meng et al., 2022, 2023, De Cao et al., 2021, Mitchell et al., 2021, 2022, Dai et al., 2021]. This ability raises natural questions: *How robust is fact retrieval, and to what extent does it depend on semantic meanings within contexts? What does it reveal about memory in LLMs?*

In this chapter, we first demonstrate that fact retrieval is not robust and LLMs can be easily fooled by varying contexts. For example, when asked to complete "The Eiffel Tower is in the city of", GPT-2 [Radford et al., 2019] answers with "Paris". However, when prompted with "The Eiffel Tower is not in Chicago. The Eiffel Tower is in the city of", GPT-2 responds with "Chicago". See Figure 4.1 for more examples, including Gemma and LLaMA. On the other hand, humans do not find the two sentences factually confusing and would answer "Paris" in both cases. We call this phenomenon *context hijacking*. Importantly, these findings suggest that LLMs might behave like an associative memory model. Specifically, we refer to an associative memory model in which LLMs rely on certain tokens in contexts to guide the retrieval of memories, even if such associations formed are not inherently semantically meaningful. This contrasts with the ideal behavior, where LLMs would generalize by understanding new contexts, reasoning through them, and integrating prior knowledge.

This associative memory perspective raises further interpretability questions about how LLMs form such associations. Answering these questions can facilitate the development of more robust LLMs. Unlike classical models of associative memory in which distance between memory patterns are measured directly and the associations between inputs and outputs are well-specified, fact retrieval relies on a more nuanced notion of similarity measured by latent (unobserved) semantic concepts. To model this, we propose a synthetic task called *latent concept association* where the output token is closely related to sampled tokens in the context but wherein similarity is measured via a latent space of semantic concepts. We then investigate how a one-layer transformer [Vaswani et al., 2017], a fundamental component of LLMs, can tackle this memory retrieval task in which various context distributions correspond to distinct memory patterns. We demonstrate that the transformer accomplishes the task in two stages: The self-attention layer gathers information, while the value matrix functions as associative memory. Moreover, low-rank structure also emerges in the embedding space of trained transformers. These findings provide additional theoretical validation for numerous existing low-rank editing and fine-tuning techniques [Meng et al., 2022, Hu et al., 2021].

Contributions Specifically, we make the following contributions:

1. We systematically demonstrate context hijacking for various open source LLM models including GPT-2 [Radford et al., 2019], LLaMA-2 [Touvron et al., 2023] and Gemma [Team et al., 2024], which show that fact retrieval can be misled by contexts (Section 4.2), reaffirming that LLMs lack robustness to context changes [Shi et al., 2023, Petroni et al., 2020, Creswell et al., 2022, Yoran et al., 2023, Pandia and Ettinger, 2021].

2. We propose a synthetic memory retrieval task termed latent concept association, allowing us to analyze how transformers can accomplish memory recall (Section 4.3). Unlike classical models of associative memory, our task creates associations in a latent,

Context Hijacking

Model	Context	Next token
All models	The Eiffel Tower is in the city of	Paris
GPT-2 / Gemma-2B	The Eiffel Tower is not in Chicago. Therefore, the Eiffel Tower is in the city of	Chicago
Gemma-2B-IT	The Eiffel Tower is not in Chicago. However, the Chicago river is in Chicago. Therefore, the Eiffel Tower is in the city of	Chicago
LLaMA-7B	The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. The Eiffel Tower is not in Chicago. Therefore, the Eiffel Tower is in the city of	Chicago

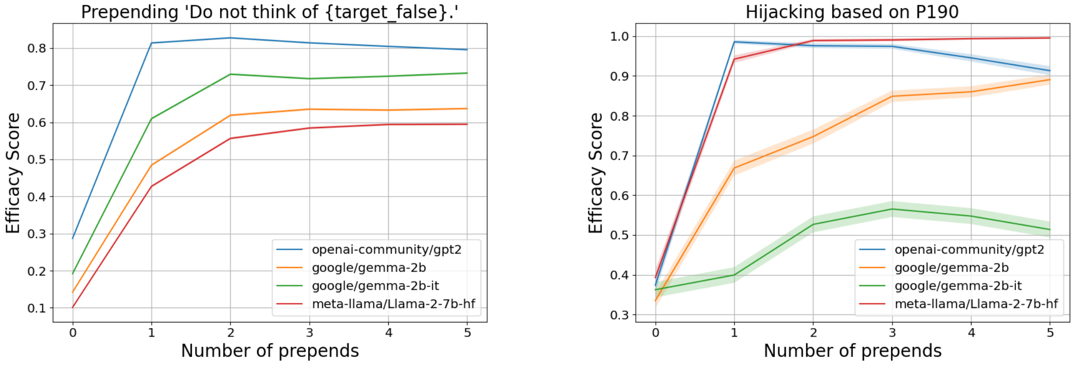
Figure 4.1: Examples of context hijacking for various LLMs, showcasing that fact retrieval is not robust.

semantic concept space as opposed to directly between observed tokens. This perspective is crucial to understanding how transformers can solve fact retrieval problems by implementing associative memory based on similarity in the latent space.

3. We theoretically (Section 4.4) and empirically (Section 4.5) study trained transformers on this latent concept association problem, showing that self-attention is used to aggregate information while the value matrix serves as associative memory. And moreover, we discover that the embedding space can exhibit a low-rank structure, offering additional support for existing editing and fine-tuning methods [Meng et al., 2022, Hu et al., 2021].

4.2 Context hijacking in LLMs

In this section, we run experiments on LLMs including GPT-2 [Radford et al., 2019], Gemma [Team et al., 2024] (both base and instruct models) and LLaMA-2-7B [Touvron et al., 2023] to explore the effects of context hijacking on manipulating LLM outputs. As an example,



(a) Hijacking generically

(b) Hijacking based on Relation ID P190

Figure 4.2: Context hijacking can cause LLMs to output false target. The figure shows efficacy score versus the number of prepends for various LLMs on the CounterFact dataset under two hijacking schemes.

consider Figure 4.1. When we prompt the LLMs with the context "The Eiffel Tower is in the city of", all 4 LLMs output the correct answer ("Paris"). However, as we see in the example, we can actually manipulate the output of the LLMs simply by modifying the context with additional *factual* information that would not confuse a human. We call this *context-hijacking*. Due to the different capacities and capabilties of each model, the examples in Figure 4.1 use different hijacking techniques. This is most notable on LLaMA-2-7B, which is a much larger model than the others. Of course, as expected, the more sophisticated attack on LLaMA also works on GPT-2 and Gemma. Additionally, the instruction-tuned version of Gemma can understand special words like "not" to some extent. Nevertheless, it is still possible to systematically hijack these LLMs, as demonstrated below.

We explore this phenomenon at scale with the CounterFact dataset introduced in Meng et al. [2022], a dataset of difficult counterfactual assertions containing a diverse set of subjects, relations, and linguistic variations. CounterFact has 21,919 samples, each of which are given by a tuple $(p, o_*, o_\_, s, r)$. From each sample, we have a context prompt p with a true target answer o_* (target_true) and a false target answer $o_\_$ (target_false), e.g. the prompt $p =$ "Eiffel Tower can be found in" has true target $o_* =$ "Paris" and false target $o_\_ =$ "Guam". Additionally, the main entity in p is the subject s ($s =$ "Eiffel Tower") and the prompt is categorized into relations r (for instance, other samples with the same relation ID as the example above could be of the form "The location of {subject} is", "{subject} can be found in", "Where is {subject}? It is in"). For additional details on how the dataset was collected, see Meng et al. [2022].

For a hijacking scheme, we report the Efficacy Score (ES) [Meng et al., 2022], which is the proportion of samples for which the token probabilities satisfy $Pr[o\_] > Pr[o_*]$ after modifying the context, that is, the proportion of the dataset that has been successfully manipulated. We experiment with two hijacking schemes for this dataset. We first hijack by prepending the text "Do not think of {target_false}" to each context. For instance, the prompt "The Eiffel Tower is in" gets changed to "Do not think of Guam. The Eiffel Tower is in". In Figure 4.2a, we see that the efficacy score rises significantly after hijacking. Here, we prepend the hijacking sentence k times for $k = 0, \dots, 5$ where $k = 0$ yields the original prompt. We see that additional prepends increase the score further.

In the second scheme, we make use of the relation ID r to prepend factually correct sentences. For instance, one can hijack the example above to "The Eiffel Tower is not located in Guam. The Eiffel Tower is in". We test this hijacking philosophy on different relation IDs. In particular, Figure 4.2b reports hijacking based on relation ID $P190$ ("twin city"). And we see similar patterns that with more prepends, the ES score gets higher. It is also worth noting that one can even hijack by only including words that are semantically close to the false target (e.g., "France" for false target "French"). This suggests that context hijacking is more than simply the LLM copying tokens from contexts. Additional details and experiments for both hijacking schemes and for other relation IDs are in Appendix C.2.

These experiments show that context hijacking changes the behavior of LLMs, leading them to output incorrect tokens, without altering the factual meaning of the context. It is worth noting that similar fragile behaviors of LLMs have been observed in the literature in different contexts [Shi et al., 2023, Petroni et al., 2020, Creswell et al., 2022, Yoran et al., 2023, Pandia and Ettinger, 2021]. See Section 4.6 for more details.

Context hijacking indicates that fact retrieval in LLMs is not robust and that accurate fact recall does not necessarily depend on the semantics of the context. As a result, one hypothesis is to view LLMs as an associative memory model where special tokens in contexts, associated with the fact, provide partial information or clues to facilitate memory retrieval [Zhao, 2023]. To better understand this perspective, we design a synthetic memory retrieval task to evaluate how the building blocks of LLMs, transformers, can solve it.

4.3 Problem setup

In the context of LLMs, fact or memory retrieval, can be modeled as a next token prediction problem. Given a context (e.g., "The capital of France is"), the objective is to accurately predict the next token (e.g., "Paris") based on the factual relation between context and the following token.

Previous papers [Ramsauer et al., 2020, Millidge et al., 2022, Bricken and Pehlevan, 2021, Zhao, 2023] have studied the connection between attention and autoassociative and heteroassociative memory. For autoassociative memory, contexts are modeled as a set of existing memories and the goal of self-attention is to select the closest one or approximations to it. On top of this, heteroassociative memory [Millidge et al., 2022, Bricken and Pehlevan, 2021] has an additional projection to remap each output to a different one, whether within the same space or otherwise. In both scenarios, the goal is to locate the closest pattern within the context when provided with a query (up to a remapping if it's heteroassociative).

Fact retrieval, on the other hand, does not strictly follow this framework. The crux of the issue is that the output token is not necessarily close to any particular token in the context but rather a combination of them and the "closeness" is intuitively measured by latent semantic concepts. For example, consider context sentence "The capital of France is" with the output "Paris". Here, none of the tokens in the context directly corresponds to the word “Paris”. Yet some tokens contain partial information about “Paris”. Intuitively, “capital” aligns with the “isCapital” concept of “Paris”, while “France” corresponds to the “isFrench” concept linked to “Paris” where all the concepts are latent. To model such phenomenon, we propose a synthetic task called *latent concept association* where the output token is closely related to tokens in the context and similarity is measured via the latent space.

4.3.1 Latent concept association

We propose a synthetic prediction task where for each output token y, tokens in the context (denoted by x) are sampled from a conditional distribution given y. Tokens that are similar to y will be favored to appear more in the context, except for y itself. The task of latent concept association is to successfully retrieve the token y given samples from $p(x|y)$. The synthetic setup simplifies by not accounting for the sequential nature of language, a choice supported by previous experiments on context hijacking (Section 4.2). We formalize this task below.

To measure similarity, we define a latent space. Here, the latent space is a collection of m binary latent variables Z_i. These could be viewed as semantic concept variables. Let $Z = (Z_1, ..., Z_m)$ be the corresponding random vector, z be its realization, and $\mathcal{Z}$ be the collection of all latent binary vectors. For each latent vector z, there’s one associated token $t \in [V] = \{0, ..., V - 1\}$ where V is the total number of tokens. Here we represent the tokenizer as ι where $\iota(z) = t$. In this chapter, we assume that ι is the standard tokenizer where each binary vector is mapped to its decimal number. In other words, there’s a one to one map between latent vectors and tokens. Because the map is one to one, we sometimes use latent vectors and tokens interchangeably. We also assume that every latent binary vector has a unique corresponding token, therefore $V = 2^m$.

Under the latent concept association model, the goal is to retrieve specific output tokens given partial information in the contexts. This is modeled by the latent conditional distribution:

$$p(z|z^*) = \omega\pi(z|z^*) + (1-\omega)\text{Unif}(\mathcal{Z})$$

where

$$\pi(z|z^*) \propto \begin{cases} \exp(-D_H(z,z^*)/\beta) & z \in \mathcal{N}(z^*), \\ 0 & z \notin \mathcal{N}(z^*). \end{cases}$$

Here D_H is the Hamming distance, $\mathcal{N}(z^*)$ is a subset of $\mathcal{Z}\backslash\{z^*\}$ and $\beta > 0$ is the temperature parameter. The use of Hamming distance draws a parallel with the notion of distributional semantics in natural language: "a word is characterized by the company it keeps" [Firth, 1957]. In words, $p(z|z^*)$ says that with probability $1-\omega$, the conditional distribution uniformly generate random latent vectors and with probability ω, the latent vector is generated from the *informative conditional distribution* $\pi(z|z^*)$ where the support of the conditional distribution is $\mathcal{N}(z^*)$. Here, π represents the informative conditional distribution that depends on z^* whereas the uniform distribution is uninformative and can be considered as noise. The mixture model parameter ω determines the signal to noise ratio of the contexts.

Therefore, for any latent vector z^* and its associated token, one can generate L context token words with the aforementioned latent conditional distribution:

- Uniformly sample a latent vector z^*
- For $l = 1, ..., L-1$, sample $z_l \sim p(z|z^*)$ and $t_l = \iota(z_l)$.
- For $l = L$, sample $z \sim \pi(z|z^*)$ and $t_L = \iota(z)$.

Consequently, we have $x = (t_1, .., t_L)$ and $y = \iota(z^*)$. The last token in the context is generated specifically to make sure that it is not from the uniform distribution. This ensures that the last token can use attention to look for clues, relevant to the output, in the context. Let $\mathcal{D}^L$ be the sampling distribution to generate (x, y) pairs. The conditional probability of y given x is given by $p(y|x)$. With slight abuse of notation, given a token $t \in [V]$, we define

$\mathcal{N}(t) = \mathcal{N}(\iota^{-1}(t))$. we also define $D_H(t, t') = D_H(\iota^{-1}(t), \iota^{-1}(t'))$ for any pair of tokens t and t'.

For any function f that maps the context to estimated logits of output labels, the training objective is to minimize this loss of the last position:

$$\mathbb{E}_{(x,y)\in\mathcal{D}^L}[\ell(f(x), y)]$$

where ℓ is the cross entropy loss with softmax. The error rate of latent concept association is defined by the following:

$$R_{\mathcal{D}^L}(f) = \mathbb{P}_{(x,y)\sim\mathcal{D}^L}[\operatorname{argmax} f(x) \neq y]$$

And the accuracy is $1 - R_{\mathcal{D}^L}(f)$.

4.3.2 Transformer network architecture

Given a context $x = (t_1, .., t_L)$ which consists of L tokens, we define $X \in \{0,1\}^{V\times L}$ to be its one-hot encoding where V is the vocabulary size. Here we use χ to represent the one-hot encoding function (i.e., $\chi(x) = X$). Similar to [Li et al., 2023b, Tarzanagh et al., 2023a, Li et al., 2024], we also consider a simplified one-layer transformer model without residual connections and normalization:

$$f^L(x) = \left[{W_E}^T W_V \operatorname{attn}(W_E \chi(x))\right]_{:L} \tag{4.3.1}$$

where

$$\operatorname{attn}(U) = U\sigma\Big(\frac{(W_K U)^T (W_Q U)}{\sqrt{d_a}}\Big),$$

$W_K \in \mathbb{R}^{d_a \times d}$ is the key matrix, and $W_Q \in \mathbb{R}^{d_a \times d}$ is the query matrix and d_a is the attention head size. $\sigma : \mathbb{R}^{L\times L} \to (0,1)^{L\times L}$ is the column-wise softmax operation. $W_V \in \mathbb{R}^{d\times d}$ is the value matrix and $W_E \in \mathbb{R}^{d\times V}$ is the embedding matrix. Here, we adopt the weight tie-in implementation which is used for Gemma [Team et al., 2024]. We focus solely on the prediction of the last position, as it is the only one relevant for latent concept association. For convenience, we also use $h(x)$ to mean $\left[\mathrm{attn}(W_E \chi(x))\right]_{:L}$, which is the hidden representation after attention for the last position, and $f_t^L(x)$ to represent the logit for output token t.

4.4 Theoretical analysis

In this section, we theoretically investigate how a single-layer transformer can solve the latent concept association problem. We first introduce a hypothetical associative memory model that utilizes self-attention for information aggregation and employs the value matrix for memory retrieval. This hypothetical model turns out to mirror trained transformers in experiments. We also examine the role of each individual component of the network: the value matrix, embeddings, and the attention mechanism. We validate our theoretical claims in Section 4.5.

4.4.1 Hypothetical associative memory model

In this section, we show that a simple single-layer transformer network can solve the latent concept association problem. The formal result is presented below in Theorem 24; first we require a few more definitions. Let $W_E(t)$ be the t-th column of the embedding matrix W_E. In other words, this is the embedding for token t. Given a token t, define $\mathcal{N}_1(t)$ to be the subset of tokens whose latent vectors are only 1 Hamming distance away from t's latent vector: $\mathcal{N}_1(t) = \{t' : D_H(t',t)) = 1\} \cap \mathcal{N}(t)$. For any output token t, $\mathcal{N}_1(t)$ contains tokens with the highest probabilities to appear in the context.

The following theorem formalizes the intuition that a one-layer transformer that uses self-attention to summarize statistics about the context distributions and whose value matrix uses aggregated representations to retrieve output tokens can solve the latent concept association problem defined in Section 4.3.1.

Theorem 24 (informal)**.** *Suppose the data generating process follows Section 4.3.1 where $m \geq 3$, $\omega = 1$, and $\mathcal{N}(t) = V \setminus \{t\}$. Then for any $\varepsilon > 0$, there exists a transformer model given by (4.3.1) that achieves error ε, i.e. $R_{\mathcal{D}^L}(f^L) < \varepsilon$ given sufficiently large context length L.*

More precisely, for the transformer in Theorem 24, we will have $W_K = 0$ and $W_Q = 0$. Each row of W_E is orthogonal to each other and normalized. And W_V is given by

$$W_V = \sum_{t \in [V]} W_E(t) (\sum_{t' \in \mathcal{N}_1(t)} W_E(t')^T) \tag{4.4.1}$$

A more formal statement of the theorem and its proof is given in Appendix C.1 (Theorem 37).

Intuitively, Theorem 24 suggests having more samples from $p(x|y)$ can lead to a better recall rate. On the other hand, if contexts are modified to contain more samples from $p(x|\tilde{y})$ where $\tilde{y} \neq y$, then it is likely for transformer to output the wrong token. This is similar to context hijacking (see Section 4.4.5). The construction of the value matrix is similar to the associative memory model used in Bietti et al. [2024], Cabannes et al. [2024], but in our case, there is no explicit one-to-one input and output pairs stored as memories. Rather, a combination of inputs are mapped to a single output.

While the construction in Theorem 24 is just one way that a single-layer transformer can tackle this task, it turns out empirically this construction of W_V is close to the trained W_V, even in the noisy case ($\omega \neq 1$). In Section 4.5.1, we will demonstrate that substituting trained value matrices with constructed ones can retain accuracy, and the constructed and trained value matrices even share close low-rank approximations. Moreover, in this hypothetical model, a simple uniform attention mechanism is deployed to allow self-attention to count occurrences of each individual tokens. Since the embeddings are orthonormal vectors, there is no interference. Hence, the self-attention layer can be viewed as aggregating information of contexts. It is worth noting that, in different settings, more sophisticated embedding structures and attention patterns are needed. This is discussed in the following sections.

4.4.2 On the role of the value matrix

The construction in Theorem 24 relies on the value matrix acting as associative memory. But is it necessary? Could we integrate the functionality of the value matrix into the self-attention module to solve the latent concept association problem? Empirically, the answer seems to be negative as will be shown in Section 4.5.1. In particular, when the context length is small, setting the value matrix to be the identity would lead to subpar memory recall accuracy.

This is because if the value matrix is the identity, the transformer would be more susceptible to the noise in the context. To see this, notice that given any pair of context and output token (x, y), the latent representation after self-attention $h(x)$ must live in the polyhedron S_y to be classified correctly where S_y is defined as:

$$S_y = \{v : (W_E(y) - W_E(t))^T v > 0 \text{ where } t \notin [V] \setminus \{y\}\}$$

Note that, by definition, for any two tokens y and $\tilde{y}$, $S_y \cap S_{\tilde{y}} = \emptyset$. On the other hand, because of the self-attention mechanism, $h(x)$ must also live in the convex hull of all the embedding vectors:

$$CV = \text{Conv}(W^E(0), ..., W^E(|V| - 1))$$

In other words, for any pair (x, y) to be classified correctly, $h(x)$ must live in the intersection of S_y and CV. Due to the stochastic nature of x, it is likely for $h(x)$ to be outside of this intersection. The remapping effect of the value matrix can help with this problem. The following lemma explains this intuition.

Lemma 25. *Suppose the data generating process follows Section 4.3.1 where* $m \geq 3$, $\omega = 1$ *and* $\mathcal{N}(t) = \{t' : D_H(t, t')) = 1\}$. *For any single layer transformer given by (4.3.1) where each row of* W_E *is orthogonal to each other and normalized, if* W_V *is constructed as in (4.4.1), then the error rate is* 0. *If* W_V *is the identity matrix, then the error rate is strictly larger than* 0.

Another intriguing phenomenon occurs when the value matrix is the identity matrix. In this case, the inner product between embeddings and their corresponding Hamming distance varies linearly. This relationship can be formalized by the following theorem.

Theorem 26. *Suppose the data generating process follows Section 4.3.1 where* $m \geq 3$, $\omega = 1$ *and* $\mathcal{N}(t) = V \setminus \{t\}$. *For any single layer transformer given by (4.3.1) with* W_V *being the identity matrix, if the cross entropy loss is minimized so that for any sampled pair* (x, y),

$$p(y|x) = \hat{p}(y|x) = softmax(f_y^L(x))$$

there exists $a > 0$ *and* b *such that for two tokens* $t \neq t'$,

$$\langle W_E(t), W_E(t') \rangle = -aD_H(t, t') + b$$

4.4.3 Embedding training and geometry

The hypothetical model in Section 4.4.1 requires embeddings to form an orthonormal basis. In the overparameterization regime where the embedding dimension d is larger than the number of tokens V, this can be approximately achieved by Gaussian initialization. However, in practice, the embedding dimension is typically smaller than the vocabulary size, in which case it is impossible for the embeddings to constitute such a basis. Empirically, in Section 4.5.2, we observe that with overparameterization ($d > V$), embeddings can be frozen at their Gaussian initialization, whereas in the underparameterized regime, embedding training is required to achieve better recall accuracy.

This raises the question: What kind of embedding geometry is learned in the underparameterized regime? Experiments reveal a close relationship between the inner product of embeddings for two tokens and the Hamming distance of these tokens (see Figure 4.3b and Figure C.3.5 in Appendix C.3.2). Approximately, we have the following relationship:

$$\langle W_E(t), W_E(t') \rangle = \begin{cases} b_0 & t = t' \\ -aD_H(t, t') + b & t \neq t' \end{cases} \tag{4.4.2}$$

for any two tokens t and t' where $b_0 > b$ and $a > 0$. One can view this as a combination of the embedding geometry under Gaussian initialization and the geometry when W_V is the identity matrix (Theorem 26). Importantly, this structure demonstrates that trained embeddings inherently capture similarity within the latent space. Theoretically, this embedding structure (4.4.2) can also lead to low error rate under specific conditions on b_0, b and a, which is articulated by the following theorem.

Theorem 27 (Informal)**.** *Following the same setup as in Theorem 24, but embeddings obey (4.4.2), then under certain conditions on a, b and if b_0 and context length L are sufficiently large, the error rate can be arbitrarily small, i.e. $R_{\mathcal{D}^L}(f^L) < \varepsilon$ for any $0 < \varepsilon < 1$.*

The formal statement of the theorem and its proof is given in Appendix C.1 (Theorem 38).

Notably, this embedding geometry also implies a low-rank structure. Let's first consider the special case when $b_0 = b$. In other words, the inner product between embeddings and their corresponding Hamming distance varies linearly.

Lemma 28. *If embeddings follow (4.4.2) and $b = b_0$ and $\mathcal{N}(t) = V \setminus \{t\}$, then rank$(W_E) \leq m + 2$.*

When $b_0 > b$, the embedding matrix will not be strictly low rank. However, it can still exhibit approximate low-rank behavior, characterized by an eigengap between the top and bottom singular values. This is verified empirically (see Figure C.3.9-C.3.12 in Appendix C.3.4).

4.4.4 The role of attention selection

As of now, attention does not play a significant role in the analysis. But perhaps unsurprisingly, the attention mechanism is useful in selecting relevant information. To see this, let's consider a specific setting where for any latent vector z^*, $\mathcal{N}(z^*) = \{z : z_1^* = z_1\} \setminus \{z^*\}$.

Essentially, latent vectors are partitioned into two clusters based on the value of the first latent variable, and the informative conditional distribution π only samples latent vectors that are in the same cluster as the output latent vector. Empirically, when trained under this setting, the attention mechanism will pay more attention to tokens within the same cluster (Section 4.5.3). This implies that the self-attention layer can mitigate noise and concentrate on the informative conditional distribution π.

To understand this more intuitively, we will study the gradient of unnormalized attention scores. In particular, the unnormalized attention score is defined as:

$$u_{t,t'} = (W_K W_E(t))^T (W_Q W_E(t'))/\sqrt{d_a}.$$

Lemma 29. *Suppose the data generating process follows Section 4.3.1 and $\mathcal{N}(z^*) = \{z : z_1^* = z_1\} \setminus \{z^*\}$. Given the last token in the sequence t_L, then*

$$\nabla_{u_{t,t_L}} \ell(f^L) = \nabla \ell(f^L)^T (W_E)^T W^V (\alpha_t \hat{p}_t W_E(t) - \hat{p}_t \sum_{l=1}^{L} \hat{p}_{t_l} W_E(t_l))$$

where for token t, $\alpha_t = \sum_{l=1}^{L} \mathbf{1}[t_l = t]$ and $\hat{p}_t$ is the normalized attention score for token t.

Typically, α_t is larger when token t and t_L belong to the same cluster because tokens within the same cluster tend to co-occur frequently. As a result, the gradient contribution to the unnormalized attention score is usually larger for tokens within the same cluster.

4.4.5 Context hijacking and the misclassification of memory recall

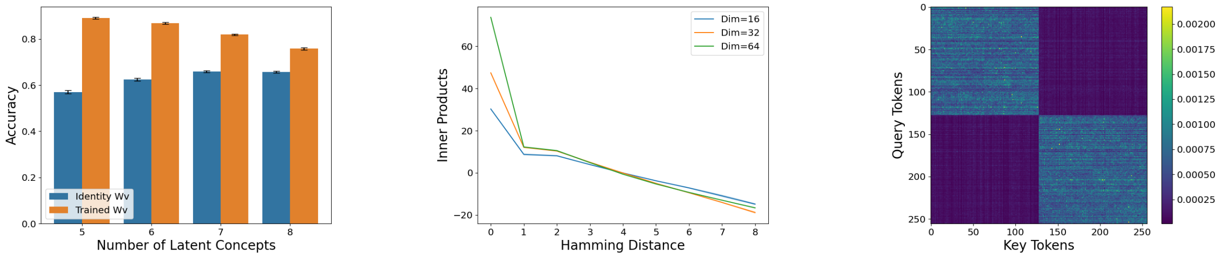
In light of the theoretical results on latent concept association, a natural question arises: How do these results connect to context hijacking in LLMs? In essence, for the latent concept association problem, the differentiation of output tokens is achieved by distinguishing between the various conditional distributions $p(x|y)$. Thus, adding or changing tokens in the context x so that it resembles a different conditional distribution can result in misclassification. In Appendix C.3.5, we present experiments showing that mixing different contexts can cause transformers to misclassify. This partially explains context hijacking in LLMs (Section 4.2). On the other hand, it is well-known that the error rate is related to the KL divergence between conditional distributions of contexts [Cover, 1999]. The closer the distributions are, the easier it is for the model to misclassify. Here, longer contexts, primarily composed of i.i.d samples, suggest larger divergences, thus higher memory recall rate. This is theoretically implied by Theorem 24 and Theorem 27 and empirically verified in Appendix C.3.6. Such result is also related to reverse context hijacking (Appendix C.2) where prepending sentences including true target words can improve fact recall rate.

4.5 Experiments

The main implications of the theoretical results in the previous section are:

1. The value matrix is important and has associative memory structure as in (4.4.1).

2. Training embeddings is crucial in the underparameterized regime, where embeddings exhibit certain geometric structures.



(a) Value matrix training

(b) Embedding structure

(c) Attention Pattern

Figure 4.3: Key components of the single-layer transformer working together on the latent concept association problem. (a) Fixing the value matrix W_V as the identity matrix results in lower accuracy compared to training W_V. The figure reports average accuracy for both fixed and trained W_V with $L = 64$. (b) When training in the underparameterized regime, the embedding structure is approximated by (4.4.2). The graph displays the average inner product between embeddings of two tokens against the corresponding Hamming distance between these tokens when $m = 8$. (c) The self-attention layer can select tokens within the same cluster. The figure shows average attention score heat map with $m = 8$ and the cluster structure from Section 4.4.4.

3. Attention mechanism is used to select the most relevant tokens.

To evaluate these claims, we conduct several experiments on synthetic datasets. Additional experimental details and results can be found in Appendix C.3.

4.5.1 On the value matrix W_V

In this section, we study the necessity of the value matrix W_V and its structure. First, we conduct experiments to compare the effects of training versus freezing W_V as the identity matrix, with the context lengths L set to 64 and 128. Figure 4.3a and Figure C.3.1 show that when the context length is small, freezing W_V can lead to a significant decline in accuracy. This is inline with Lemma 25 and validates it in a general setting, implying the significance of the value matrix in maintaining a high memory recall rate.

Next, we investigate the degree of alignment between the trained value matrix W_V and the construction in (4.4.1). The first set of experiments examines the similarity in functionality between the two matrices. We replace value matrices in trained transformers with the constructed ones like in (4.4.1) and then report accuracy with the new value matrix. As a baseline, we also consider randomly constructed value matrix, where the outer product pairs are chosen randomly (detailed construction can be found in Appendix C.3.1). Figure C.3.2 indicates that the accuracy does not significantly decrease when the value matrix is replaced with the constructed ones. Furthermore, not only are the constructed value matrix and the trained value matrix functionally alike, but they also share similar low-rank approximations. We use singular value decomposition to get the best low rank approximations of various value matrices where the rank is set to be the same as the number of latent variables (m). We then compute smallest principal angles between low-rank approximations of trained value matrices and those of constructed, randomly constructed, and Gaussian-initialized value matrices. Figure C.3.3 shows that the constructed ones have, on average, smallest principal angles with the trained ones.

4.5.2 On the embeddings

In this section, we explore the significance of embedding training in the underparamerized regime and embedding structures. We conduct experiments to compare the effects of training versus freezing embeddings with different embedding dimensions. The learning rate is selected as the best option from $\{0.01, 0.001\}$ depending on the dimensions. Figure C.3.4 clearly shows that when the dimension is smaller than the vocabulary size ($d < V$), embedding training is required. It is not necessary in the overparameterized regime ($d > V$), partially confirming Theorem 24 because if embeddings are initialized from a high-dimensional multi-variate Gaussian, they are approximately orthogonal to each other and have the same norms.

The next question is what kind of embedding structures are formed for trained transformers in the underparamerized regime. From Figure 4.3b and Figure C.3.5, it is evident that the relationship between the average inner product of embeddings for two tokens and their corresponding Hamming distance roughly aligns with (4.4.2). Perhaps surprisingly, if we plot the same graph for trained transformers with a fixed identity value matrix, the relationship is mostly linear as shown in Figure C.3.6, confirming our theory (Theorem 26).

As suggested in Section 4.4.3, such embedding geometry (4.4.2) can lead to low rank structures. We verify this claim by studying the spectrum of the embedding matrix W_E. As illustrated in Appendix C.3.4, Figure C.3.9-C.3.12 demonstrate that there are eigengaps between top and bottom singular values, suggesting low-rank structures.

4.5.3 On the attention selection mechanism

In this section, we examine the role of attention pattern by considering a special class of latent concept association model as defined in Section 4.4.4. Figure 4.3c and Figure C.3.7 clearly show that the self-attention select tokens in the same clusters. This suggests that attention can filter out noise and focus on the informative conditional distribution π. We extend experiments to consider cluster structures that depend on the first two latent variables (detailed construction can be found in Appendix C.3.3) and Figure C.3.8 shows attention pattern as expected.

4.6 Related Work

Associative memory Associative memory has been explored within the field of neuroscience [Hopfield, 1982, Seung, 1996, Ben-Yishai et al., 1995, Skaggs et al., 1994, Steinberg and Sompolinsky, 2022]. The most popular models among them is the Hopfield network [Hopfield, 1982] and its modern successors [Ramsauer et al., 2020, Millidge et al., 2022, Zhao, 2023, Hu et al., 2024d, Wu et al., 2023, Hu et al., 2024b,c, Wu et al., 2024a, Hu et al., 2024a] are closely related to the attention layer used in transformers [Vaswani et al., 2017]. In addition, the attention mechanism has also been shown to approximate another associative memory model known as sparse distributed memory [Bricken and Pehlevan, 2021]. Beyond attention, Radhakrishnan et al. [2020], Jiang and Pehlevan [2020] show that overparameterzed autoencoders can implement associative memory as well. This work studies fact retrieval as a form of associative memory. Another closely related area of research focuses on memorization in deep neural networks. Henighan et al. [2023] shows that a simple neural network trained on toy model will store data points in the overfitting regime while storing features in the underfitting regime. Feldman [2020], Feldman and Zhang [2020] study the interplay between memorization and long tail distributions while Kim et al. [2022], Mahdavi et al. [2023] study the memorization capacity of transformers.

Interpreting transformers and LLMs There's a growing body of work on understanding how transformers and LLMs work [Li et al., 2023b, Allen-Zhu and Li, 2023a,b, 2024, Emrullah Ildiz et al., 2024, Tarzanagh et al., 2023b,a, Li et al., 2024], including training dynamics [Tian et al., 2023a,b, Sheen et al., 2024] and in-context learning [Xie et al., 2021, Garg et al., 2022, Bai et al., 2024,?]. Recent papers have introduced synthetic tasks to better understand the mechanisms of transformers [Charton, 2022, Liu et al., 2022, Nanda et al., 2023a, Zhang et al., 2022, Zhong et al., 2024], such as those focused on Markov chains [Bietti et al., 2024, Edelman et al., 2024, Nichani et al., 2024, Makkuva et al., 2024]. Most notably, Bietti et al. [2024] and subsequent works [Cabannes et al., 2023, 2024] study weights in transformers as associative memory but their focus is on understanding induction head [Olsson et al., 2022b] and one-to-one map between input query and output memory. An increasing amount of research is dedicated to understanding the internals of pre-trained LLMs, broadly categorized under the term "mechanistic interpretability" [Elhage et al., 2021, Olsson et al., 2022a, Geva et al., 2023, Meng et al., 2022, 2023, Jiang et al., 2024b, Rajendran et al., 2024b, Hase et al., 2024, Wang et al., 2022, McGrath et al., 2023, Geiger et al., 2021, 2022, 2024, Wu et al., 2024b].

Knowledge editing and adversarial attacks on LLMs Fact recall and knowledge editing have been extensively studied [Meng et al., 2022, 2023, Hase et al., 2024, Sakarvadia et al., 2023, De Cao et al., 2021, Mitchell et al., 2021, 2022, Dai et al., 2021, Zhang et al., 2023, Tian et al., 2024, Jin et al., 2023], including the use of in-context learning to edit facts [Zheng et al., 2023]. This work aims to explore a different aspect by examining the robustness of fact recall to variation in prompts. A closely related line of work focuses on adversarial attacks on LLMs [see Chowdhury et al., 2024, for a review]. Specifically, prompt-based adversarial attacks [Xu et al., 2023, Zhu et al., 2023, Wang et al., 2023c] focus on the manipulation of answers within specific classification tasks while other works concentrate on safety issues [Liu et al., 2023a, Perez and Ribeiro, 2022, Zou et al., 2023, Apruzzese et al., 2022, Wang et al., 2023a, Si et al., 2022, Rao et al., 2023, Shanahan et al., 2023, Liu et al., 2023b]. Yu et al. [2024], Luo et al. [2024] also study jailbreak phenomena within the context of modern Hopfield network. There are also works showing LLMs can be distracted by irrelevant contexts in problem solving [Shi et al., 2023], question answering [Petroni et al., 2020, Creswell et al., 2022, Yoran et al., 2023] and factual reasoning [Pandia and Ettinger, 2021]. Although phenomena akin to context hijacking have been reported in different instances, the goals of this work are to give a systematic robustness study for fact retrieval, offer a framework for interpreting it in the context of associative memory, and deepen our understanding of LLMs.

4.7 Conclusions

In this work, we first presented the phenomenon of context hijacking in LLMs, which suggested that fact retrieval is not robust against variations of contexts. This indicates that LLMs might function like associative memory where tokens in contexts are clues to guide memory retrieval. To investigate this perspective further, we devised a synthetic task called latent concept association and examined theoretically and empirically how single-layer transformers are trained to solve this task. These results provide further insights into the inner workings of transformers and LLMs, and can hopefully stimulate further work into interpreting and understanding the mechanisms by which LLMs predict tokens and recall facts.

Limitations The context hijacking experiments were only conducted on open-source models and not on commercial models like GPT-4. Nevertheless, even in the official GPT-4 technical report [Achiam et al., 2023], there is an example similar to context hijacking (the Elvis Perkins example). In that example, the prompt is "Son of an actor, this American guitarist and rock singer released many songs and albums and toured with his band. His name is "Elvis" what?". GPT-4 answers with Presley, even though the answer is Perkins (Elvis Presley is not the son of an actor). GPT-4 can be viewed as distracted by all the information related to music and answers Presley. In fact, it is known that LLMs can be easily distracted by contexts in use cases other than fact retrieval such as problem-solving [Shi et al., 2023]. So we reasonably suspect that similar behavior still exists in larger models but is harder to exploit. On the other hand, the theoretical section only focuses on single-layer transformer network. While single-layer networks already demonstrate some interesting phenomena including low-rank structures, the functionality of multi-layer transformers is much different compared to single-layer transformers with the notable emergence of induction head [Elhage et al., 2021].

CHAPTER 5

LOOKING FORWARD

This dissertation examines the representations learned by foundation models and demonstrates that—even as mere "projections" of reality—they are capable of encoding the world's underlying structures and relationships. Specifically, we show that linear representations capture the binary contrast between counterfactual tokens, partial orthogonality encodes semantic independence, and self-attention can leverage the structure in representations to retrieve information. It is worth noting the distinction between this line of research and causal representation learning [Schölkopf et al., 2021]: whereas causal representation learning aims to recover the latent distribution, this thesis focuses exclusively on the relationships among latent (concept) variables.

Several promising avenues remain unexplored: First, although interpreting foundation models is interesting in its own right, the most valuable applications of these insights are still unclear. It remains an open question whether interoperability-based methods truly outperform more established techniques like prompting or fine-tuning [Wu et al., 2025]. Turning interpretability results into practical tools is therefore an important, unresolved challenge. Second, this dissertation focuses primarily on token representations as a window into a model's internals, but representations need not be limited to fixed vectors. A model's beliefs or knowledge of facts might be encoded in richer or more abstract forms. Discovering and formalizing alternative abstractions offers another fruitful direction for future work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(1), 2010.

Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. In *International Conference on Machine Learning*, pages 223–231. PMLR, 2019.

Carl Allen, Ivana Balazevic, and Timothy Hospedales. What the vec? towards probabilistically grounded embeddings. *Advances in neural information processing systems*, 32, 2019.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction, 2023a.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation, 2023b.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws, 2024.

Arash A Amini, Bryon Aragam, and Qing Zhou. A non-graphical representation of conditional independence via the neighbourhood lattice. *arXiv preprint arXiv:2206.05829*, 2022.

Giovanni Apruzzese, Hyrum S. Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin A. Roundy. "real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice, 2022.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David M. Mimno, Ankur Moitra, David A. Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 280–288. JMLR.org, 2013.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *arXiv preprint arXiv:1502.03520*, pages 385–399, 2015.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

Rani Ben-Yishai, R Lev Bar-Or, and Haim Sompolinsky. Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences*, 92(9):3844–3848, 1995.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36, 2024.

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120, 2006.

Tobias Boege and Thomas Kahle. Construction methods for gaussoids. *arXiv preprint arXiv:1902.11260*, 2019.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Trenton Bricken and Cengiz Pehlevan. Attention approximates sparse distributed memory. *Advances in Neural Information Processing Systems*, 34:15301–15315, 2021.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. *arXiv preprint arXiv:2306.02235*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories. *arXiv preprint arXiv:2310.02984*, 2023.

Vivien Cabannes, Berfin Simsek, and Alberto Bietti. Learning associative memories with gradient descent. *arXiv preprint arXiv:2402.18724*, 2024.

Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. The geometry of multilingual language model representations. *arXiv preprint arXiv:2205.10964*, 2022.

François Charton. What is my math transformer doing?–three results on interpretability and generalization. *arXiv preprint arXiv:2211.00170*, 2022.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. Probing bert in hyperbolic spaces. *arXiv preprint arXiv:2104.03869*, 2021.

Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vinija Jain, and Aman Chadha. Breaking down the defenses: A comparative survey of attacks on large language models. *arXiv preprint arXiv:2403.04786*, 2024.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

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

Sanjoy Dasgupta. Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 634–644. IEEE, 1999.

A Philip Dawid. Separoids: A mathematical framework for conditional independence and irrelevance. *Annals of Mathematics and Artificial Intelligence*, 32:335–372, 2001.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

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.

Luc Devroye. The equivalence of weak, strong and complete convergence in l1 for kernel density estimates. *The Annals of Statistics*, 11(3):896–904, 1983. ISSN 00905364. URL `http://www.jstor.org/stable/2240651`.

Benjamin L Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains. *arXiv preprint arXiv:2402.11004*, 2024.

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

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022a.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022b.

M Emrullah Ildiz, Yixiao Huang, Yingcong Li, Ankit Singh Rawat, and Samet Oymak. From self-attention to markov models: Unveiling the dynamics of generative transformers. *arXiv e-prints*, pages arXiv–2402, 2024.

Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*, 2017.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*, 2018.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies, 2019.

Fabian Falck, Haoting Zhang, Matthew Willetts, George Nicholson, Christopher Yau, and Chris C Holmes. Multi-facet clustering variational autoencoders. *Advances in Neural Information Processing Systems*, 34, 2021.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Alex Fink. The binomial ideal of the intersection axiom for conditional probabilities. *Journal of Algebraic Combinatorics*, 33:455–463, 2011.

John Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32, 1957.

W Nelson Francis and Henry Kucera. Brown corpus manual. *Letters to the Editor*, 5(2):7, 1979.

Abraham Frandsen and Rong Ge. Understanding composition of word embeddings via tensor decomposition. *arXiv preprint arXiv:1902.00613*, 2019.

Ming Gao and Bryon Aragam. Efficient bayesian network structure learning via local markov boundary search. *Advances in Neural Information Processing Systems*, 34:4301–4313, 2021.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

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

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR, 2024.

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

Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. Skip-gram - zipf + uniform = vector additivity. In *Annual Meeting of the Association for Computational Linguistics*, 2017a.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, 2017b.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, 2016.

Daniel Greenfeld and Uri Shalit. Robust learning with the hilbert-schmidt independence criterion. In *International Conference on Machine Learning*, pages 3759–3768. PMLR, 2020.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory: 16th International Conference, ALT 2005, Singapore, October 8-11, 2005. Proceedings 16*, pages 63–77. Springer, 2005.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Tom Henighan, Shan Carter, Tristan Hume, Nelson Elhage, Robert Lasenby, Stanislav Fort, Nicholas Schiefer, and Christopher Olah. Superposition, memorization, and double descent. *Transformer Circuits Thread*, 2023.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Robin Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. *arXiv preprint arXiv:2404.03828*, 2024a.

Jerry Yao-Chieh Hu, Bo-Yu Chen, Dennis Wu, Feng Ruan, and Han Liu. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024b.

Jerry Yao-Chieh Hu, Thomas Lin, Zhao Song, and Han Liu. On computational limits of modern hopfield models: A fine-grained complexity analysis. *arXiv preprint arXiv:2402.04520*, 2024c.

Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. *Advances in Neural Information Processing Systems*, 36, 2024d.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024.

Aapo Hyvärinen, Ilyes Khemakhem, and Ricardo Monti. Identifiability of latent-variable and structural-equation models: from linear to nonlinear. *arXiv preprint arXiv:2302.02672*, 2023.

Ilse CF Ipsen and Rizwana Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776, 2008.

Yibo Jiang and Bryon Aragam. Learning latent causal graphs with unknown interventions. In *Advances in Neural Information Processing Systems*, 2023.

Yibo Jiang and Cengiz Pehlevan. Associative memory in iterated overparameterized sigmoid autoencoders. In *International conference on machine learning*, pages 4828–4838. PMLR, 2020.

Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *Advances in Neural Information Processing Systems*, 35:20782–20794, 2022.

Yibo Jiang, Bryon Aragam, and Victor Veitch. Uncovering meanings of embeddings via partial orthogonality. *arXiv preprint arXiv:2310.17611*, 2023.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Do llms dream of elephants (when told not to)? latent concept association and associative memory in transformers. *Advances in Neural Information Processing Systems*, 37:67712–67757, 2024a.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. 2024b.

Tian Jin, Nolan Clement, Xin Dong, Vaishnavh Nagarajan, Michael Carbin, Jonathan Ragan-Kelley, and Gintare Karolina Dziugaite. The cost of down-scaling language models: Fact recall deteriorates before in-context learning. *arXiv preprint arXiv:2310.04680*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

Junghwan Kim, Michelle Kim, and Barzan Mozafari. Provable memorization capacity of transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Learning latent causal graphs via mixture oracles. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18087–18101, 2021.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022.

Andrew V Knyazev and Merico E Argentati. Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Sébastien Lachapelle, Pau Rodríguez, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *1st Conference on Causal Learning and Reasoning, CLeaR 2022, Sequoia Conference Center, Eureka, CA, USA, 11-13 April, 2022*, volume 177 of *Proceedings of Machine Learning Research*, pages 428–484. PMLR, 2022.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Steffen L Lauritzen. *Lectures on graphical models.* 2020. URL `http://web.math.ku.dk/~{}lauritzen/papers/gmnotes.pdf`.

Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci. When are post-hoc conceptual explanations identifiable? In *Uncertainty in Artificial Intelligence*, pages 1207–1218. PMLR, 2023.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.* Citeseer, 2012.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*, 2020.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*, 2023a.

Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pages 685–693. PMLR, 2024.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In *International Conference on Machine Learning*, pages 19689–19729. PMLR, 2023b.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023a.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023b.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

Radim Lněnička and František Matúš. On gaussian conditional independence structures. *Kybernetika*, 43(3):327–342, 2007.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Haozheng Luo, Jiahao Yu, Wenxin Zhang, Jialong Li, Jerry Yao-Chieh Hu, Xingyu Xin, and Han Liu. Decoupled alignment for robust plug-and-play adaptation. *arXiv preprint arXiv:2406.01514*, 2024.

Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. Memorization capacity of multi-head attention in transformers. *arXiv preprint arXiv:2306.02010*, 2023.

Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains, 2024.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47): e2206625119, 2022.

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

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer, 2023.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv:2305.16130*, 2023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013b.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995. ISSN 0001-0782. doi:10.1145/219717.219748. URL `https://doi.org/10.1145/219717.219748`.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pages 15561–15583. PMLR, 2022.

David Mimno and Laure Thompson. The strange geometry of skip-gram with negative sampling. In *Conference on Empirical Methods in Natural Language Processing*, 2017a.

David Mimno and Laure Thompson. The strange geometry of skip-gram with negative sampling. In *Empirical Methods in Natural Language Processing*, 2017b.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

Joris Mooij, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf. Regression by dependence minimization and its application to causal inference in additive noise models. In *Proceedings of the 26th annual international conference on machine learning*, pages 745–752, 2009.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodola. Relative representations enable zero-shot latent space communication. *arXiv preprint arXiv:2209.15430*, 2022.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023a.

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

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent, 2024.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022a.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022b.

OpenAI. GPT-4 technical report, 2023.

Lalchand Pandia and Allyson Ettinger. Sorting through the noise: Testing robustness of information processing in pre-trained language models. *arXiv preprint arXiv:2109.12393*, 2021.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2023.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. 2024.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jose M Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Pramuditha Perera, Matthew Trager, Luca Zancato, Alessandro Achille, and Stefano Soatto. Prompt algebra for task composition. *arXiv preprint arXiv:2306.00310*, 2023.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

Jonas Peters. On the intersection property of conditional independence and its application to causal discovery. *Journal of Causal Inference*, 3(1):97–108, 2015.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks implement associative memory. *Proceedings of the National Academy of Sciences*, 117(44):27162–27170, 2020.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep understanding and improvement. *stat*, 1050: 19, 2017.

Goutham Rajendran, Bohdan Kivva, Ming Gao, and Bryon Aragam. Structure learning in polynomial time: Greedy algorithms, bregman information, and exponential families. *Advances in Neural Information Processing Systems*, 34:18660–18672, 2021.

Goutham Rajendran, Patrik Reizinger, Wieland Brendel, and Pradeep Ravikumar. An interventional perspective on identifiability in gaussian lti systems with independent component analysis. *arXiv preprint arXiv:2311.18048*, 2023.

Goutham Rajendran, Simon Buchholz, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning interpretable concepts: Unifying causal representation learning and foundation models. *arXiv preprint*, 2024a.

Goutham Rajendran, Simon Buchholz, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning interpretable concepts: Unifying causal representation learning and foundation models. *arXiv preprint arXiv:2402.09236*, 2024b.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Understanding, analyzing, and preventing jailbreaks. *arXiv preprint arXiv:2305.14965*, 2023.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32, 2019.

Narutatsu Ri, Fei-Tzin Lee, and Nakul Verma. Contrastive loss is all you need to recover analogies as parallel lines. *arXiv preprint arXiv:2306.08221*, 2023.

Maja Rudolph and David Blei. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*, 2017.

Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. Exponential family embeddings. *Advances in Neural Information Processing Systems*, 29, 2016.

Kayvan Sadeghi. Faithfulness of probability distributions and graphs. *Journal of Machine Learning Research*, 18(148), 2017.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. *arXiv preprint arXiv:2309.05605*, 2023.

Ernesto San Martin, Michel Mouchart, and Jean-Marie Rolin. Ignorable common information, null sets and basu's first theorem. *Sankhyā: The Indian Journal of Statistics*, pages 674–698, 2005.

Bernhard Schölkopf and Julius von Kügelgen. From statistical to causal learning. *arXiv preprint arXiv:2204.00607*, 2022.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. arXiv:2102.11107.

Lisa Schut, Nenad Tomasev, Tom McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero. *arXiv preprint arXiv:2310.16410*, 2023.

H Sebastian Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93(23):13339–13344, 1996.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

Heejune Sheen, Siyu Chen, Tianhao Wang, and Harrison H Zhou. Implicit regularization of gradient flow on one-layer softmax attention. *arXiv preprint arXiv:2403.08699*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR, 2023.

Wai Man Si, Michael Backes, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, Savvas Zannettou, and Yang Zhang. Why so toxic? measuring and triggering toxic behavior in open-domain chatbots. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2659–2673, 2022.

William Skaggs, James Knierim, Hemant Kudrimoti, and Bruce McNaughton. A model of the neural basis of the rat's sense of direction. *Advances in neural information processing systems*, 7, 1994.

Le Song, Jonathan Huang, Alex Smola, and Kenji Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968, 2009.

Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search.* MIT press, 2000.

Chandler Squires and Caroline Uhler. Causal structure learning: a combinatorial perspective. *Foundations of Computational Mathematics*, pages 1–35, 2022.

Alexander Statnikov, Jan Lemeir, and Constantin F Aliferis. Algorithms for discovery of multiple markov boundaries. *The Journal of Machine Learning Research*, 14(1):499–566, 2013.

Julia Steinberg and Haim Sompolinsky. Associative memory of structured knowledge. *Scientific Reports*, 12(1):21808, 2022.

Eric V Strobl and Shyam Visweswaran. Markov boundary discovery with ridge regularized linear models. *Journal of Causal inference*, 4(1):31–48, 2016.

Milan Studený. *Probabilistic conditional independence structures*. Information science and statistics. Springer, 2005. ISBN 978-1-85233-891-6.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023a.

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Margin maximization in attention mechanism. *arXiv preprint arXiv:2306.13596*, 2023b.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Ningyu Zhang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: Instruction-based knowledge editing for large language models. *arXiv preprint arXiv:2402.16123*, 2024.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in Neural Information Processing Systems*, 36:71911–71947, 2023a.

Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *arXiv preprint arXiv:2310.00535*, 2023b.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al.

Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. Linear spaces of meanings: compositional structures in vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15395–15404, 2023.

Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–678, 2003a.

Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, volume 2, pages 376–380. St. Augustine, FL, 2003b.

Burak Varici, Emre Acarturk, Karthikeyan Shanmugam, Abhishek Kumar, and Ali Tajer. Score-based causal representation learning with interventions. *arXiv preprint arXiv:2301.08230*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Santosh S. Vempala. The random projection method. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2005.

Riccardo Volpi and Luigi Malagò. Evaluating natural alpha embeddings on intrinsic and extrinsic tasks. In *Workshop on Representation Learning for NLP*, 2020.

Riccardo Volpi and Luigi Malagò. Natural alpha embeddings. *Information Geometry*, 4(1): 3–29, 2021.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv preprint arXiv:2306.11698*, 2023a.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. 2023b.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023c.

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

Yue Wang and Linbo Wang. Causal inference in degenerate systems: An impossibility result. In *International Conference on Artificial Intelligence and Statistics*, pages 3383–3392. PMLR, 2020.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for score-based conditional model. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023d.

Zihao Wang, Yibo Jiang, Jiahao Yu, and Heqing Huang. The illusion of role separation: Hidden shortcuts in llm role learning (and how to fix them). 2025.

Joe Whittaker. *Graphical models in applied multivariate statistics*. Wiley Publishing, 2009.

Dennis Wu, Jerry Yao-Chieh Hu, Weijian Li, Bo-Yu Chen, and Han Liu. Stanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. *arXiv preprint arXiv:2312.17346*, 2023.

Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. *arXiv preprint arXiv:2404.03827*, 2024a.

Jingfeng Wu, Difan Zou, Vladimir Braverman, and Quanquan Gu. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. *arXiv preprint arXiv:2011.02538*, 2020.

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

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. An llm can fool itself: A prompt-based adversarial attack. *arXiv preprint arXiv:2310.13345*, 2023.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*, 2023.

Jiahao Yu, Haozheng Luo, Jerry Yao-Chieh, Wenbo Guo, Han Liu, and Xinyu Xing. Enhancing jailbreak attack against large language models through silent tokens. *arXiv preprint arXiv:2405.20653*, 2024.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task. *arXiv preprint arXiv:2206.04301*, 2022.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. How do large language models capture the ever-changing world knowledge? a review of recent advances. *arXiv preprint arXiv:2310.07343*, 2023.

Jiachen Zhao. In-context exemplars as clues to retrieving from large associative memory. *arXiv preprint arXiv:2311.03498*, 2023.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*, 2023.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*, 2023.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

APPENDIX A

APPENDIX FOR CHAPTER 2

A.1 Linearity from log-odds: Proof of Theorem 3

In this section, we prove Theorem 3 restated below for convenience.

Theorem 3 (Log-odds implies linearity)**.** *Fix a concept $i \in [m]$. Suppose for any concept vector $c \in \mathcal{C}$ and context $d \in \mathcal{D}$ such that $d_i = \diamond$, we have*

$$\ln \frac{\hat{p}(c_{(i \to 0)}|d)}{\hat{p}(c_{(i \to 1)}|d)} = \ln \frac{p(C_i = 0)}{p(C_i = 1)}$$

Then, the vectors $\overline{\Delta_{c,i}}$ over all $c \in \mathcal{C}$ are parallel.

Proof. Fix any $i \leq m$ and consider the vector $\overline{\Delta_{c,i}} = \Pi_i(g(c_{(i \to 0)}) - g(c_{(i \to 1)}))$ for any $c \in \mathcal{C}$. Let $\widehat{\mathcal{D}} = \{d \in \{\diamond, 0, 1\}^m | d_i = \diamond\}$ be the contexts that do not condition on C_i and let $u_1, \dots, u_k$ be an orthonormal basis of the space $\mathrm{span}\{f(d) | d \in \widehat{\mathcal{D}}\}$. Also, since $f(d)$ over $d \in \widehat{\mathcal{D}}$ span this space, let

$$u_j = \sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} f(d)$$

for all $j = 1, 2, \dots, k$ and scalars $\alpha_{j,d}$. Note that the $\alpha_{j,d}$ do not depend on c. Now, for any

$j \leq k$,

$$\begin{aligned}
\langle \overline{\Delta_{c,i}}, u_j \rangle &= \langle \Pi_i (g(c_{(i \to 0)}) - g(c_{(i \to 1)})), u_j \rangle \\
&= \langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), \Pi_i u_j \rangle \\
&= \langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), u_j \rangle \\
&= \langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), \sum_{d \in \mathcal{D}} \alpha_{j,d} f(d) \rangle \\
&= \sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), f(d) \rangle
\end{aligned}$$

To this end, we will compute the inner product $\langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), f(d) \rangle$ for a fixed $d \in \widehat{\mathcal{D}}$. First,

$$\ln \frac{\hat{p}(c_{(i \to 0)} | d)}{\hat{p}(c_{(i \to 1)} | d)} = \ln \frac{p(C_i = 0)}{p(C_i = 1)}$$

But we have

$$\hat{p}(c_{(i \to 0)} | d) = \frac{\exp(f(d)^T g(c_{(i \to 0)}))}{\sum_{c'} \exp(f(d)^T g(c'))}, \qquad \hat{p}(c_{(i \to 1)} | d) = \frac{\exp(f(d)^T g(c_{(i \to 1)}))}{\sum_{c'} \exp(f(d)^T g(c'))}$$

Since the denominator does not depend on c, rearranging implies

$$\begin{aligned}
\langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), f(d) \rangle &= f(d)^T (g(c_{(i \to 1)}) - g(c_{(i \to 0)})) \\
&= \ln \frac{\hat{p}(c_{(i \to 0)} | d)}{\hat{p}(c_{(i \to 1)} | d)} \\
&= \ln \frac{p(C_i = 0)}{p(C_i = 1)}
\end{aligned}$$

which only depends on i. Call this expression $\alpha^{(i)}$ to get

$$\begin{aligned}\langle \overline{\Delta_{c,i}}, u_j \rangle &= \sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), f(d) \rangle \\ &= \sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \alpha^{(i)}\end{aligned}$$

Therefore,

$$\begin{aligned}\overline{\Delta_{c,i}} &= \sum_{j \le k} \langle \overline{\Delta_{c,i}}, u_j \rangle u_j \\ &= \sum_{j \le k} \left(\sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \alpha^{(i)} \right) u_j \\ &= \alpha^{(i)} \sum_{j \le k, d \in \widehat{\mathcal{D}}} \alpha_{j,d} u_j\end{aligned}$$

Note that regardless of c, the final expression is always parallel to the vector $v_i = \sum_{j \le k, d \in \widehat{\mathcal{D}}} \alpha_{j,d} u_j$ which does not depend on c. This completes the proof. □

A.2 Linearity from log-odds for general MRFs

In this section, we generalize the ideas from Appendix A.1 to concepts from a general Markov random field, which is more general than the independent case. Since most of the technical ideas and motivations are in Appendix A.1, we will go over them lightly here. The goal is to study the structure of the steering vector $\Delta_{c,i} = g(c_{(i \to 1)}) - g(c_{(i \to 0)})$. As we will see, instead of them all lying in a space of dimension 1 as in the independent case (that's what being parallel means), they will now live in a subspace of low dimension. For example, such a phenomenon was experimentally observed by Li et al. [2023a] for the concept of *truthfulness*.

Here, concepts $C_1, \ldots, C_m$ come from a Markov random field with undirected graph $G_C = (V_C, E_C)$ with neighborhood set given by $\text{ne}(i)$. Therefore, they satisfy the property that $p(C_i|C_{[m]\backslash i}) = p(C_i|C_{\text{ne}(i)})$ for all $i \leq m$. Accordingly, we state the log-odds assumption to capture this general conditional independence.

Assumption 1. For any concept $i \leq m$, any concept vector $c \in \mathcal{C}$ and context $d \in \{\diamond, 0, 1\}^m$ such that $d_i = \diamond$ and $d_j \neq \diamond$ for all $j \in \text{ne}(i)$, we have

$$\ln \frac{\hat{p}(c_{(i\to 0)}|d)}{\hat{p}(c_{(i\to 1)}|d)} = \ln \frac{p(C_i = 0|C_{\text{ne}(i)} = d_{\text{ne}(i)})}{p(C_i = 1|C_{\text{ne}(i)} = d_{\text{ne}(i)})}$$

As before, we assume above that the log odds condition holds when i is not conditioned on, but we weaken this further to say that it only needs to hold specifically when every one of its neighbors j has been conditioned on.

Analogously, we project out the space that could possibly contribute to a 0 conditional probability. This is the space where either d_i has been conditioned on or d_j for some $j \in \text{ne}(i)$ has been conditioned on. Therefore, define $\overline{\Delta_{c,i}} = \Pi_i \Delta_{c,i}$ where Π_i is the projection into the space $\text{span}\{f(d)|d_i = \diamond, d_j \neq \diamond \forall j \in \text{ne}(i)\}$ We now state our main theorem.

Theorem 30. *Under Assumption 1, for any fixed $i \leq m$, the vectors $\overline{\Delta_{c,i}}$ for all $c \in \mathcal{C}$ live in a subspace $\mathcal{S}$ of dimension at most $2^{|ne(i)|}$.*

This theorem says that if we assume that the concepts come from a general Markov random field, then the steering vectors live in a space of low dimension. Note that when the concepts are independent, we have $|\text{ne}(i)| = 0 \implies 2^{|\text{ne}(i)|} = 1$ which means all the vectors $\overline{\Delta_{c,i}}$ are parallel. Therefore this theorem is more general than Theorem 3.

Proof. We will proceed similarly to the proof of Theorem 3. Fix any $i \leq m, c \in \mathcal{C}$ and repeat the computation until we get that for any $j \leq k$,

$$\langle \overline{\Delta_{c,i}}, u_j \rangle = \sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \langle g(c_{(i\to 0)}) - g(c_{(i\to 1)}), f(d) \rangle$$

where $\widehat{\mathcal{D}} = \{d \in \{\diamond, 0, 1\}^m | d_i = \diamond, d_j \neq \diamond \forall j \in \text{ne}(i)\}$. Now, let's recompute $\langle g(c_{(i \to 0)}) - g(c_{(i \to 1)}), f(d) \rangle$. By Assumption 1, we have

$$\ln \frac{\hat{p}(c_{(i\to 0)}|d)}{\hat{p}(c_{(i\to 1)}|d)} = \ln \frac{p(C_i = 0|C_{\text{ne}(i)} = d_{\text{ne}(i)})}{p(C_i = 1|C_{\text{ne}(i)} = d_{\text{ne}(i)})}$$

which gives

$$\langle g(c_{(i\to 0)}) - g(c_{(i\to 1)}), f(d) \rangle = \ln \frac{p(C_i = 0|C_{\text{ne}(i)} = d_{\text{ne}(i)})}{p(C_i = 1|C_{\text{ne}(i)} = d_{\text{ne}(i)})}$$

which depends both on i and $d_{\text{ne}(i)}$. For all $\sigma \in \{0,1\}^{|\text{ne}(i)|}$, denote $\alpha^{(i),\sigma}$ to be the expression

$$\alpha^{(i),\sigma} = \ln \frac{p(C_i = 0|C_{\text{ne}(i)} = \sigma)}{p(C_i = 1|C_{\text{ne}(i)} = \sigma)}$$

Therefore,

$$\begin{aligned}
\overline{\Delta_{c,i}} &= \sum_{j \leq k} \left(\sum_{d \in \widehat{\mathcal{D}}} \alpha_{j,d} \alpha^{(i),d_{\text{ne}(i)}} \right) u_j \\
&= \sum_{\sigma \in \{0,1\}^{|\text{ne}(i)|}} \alpha^{(i),\sigma} \left(\sum_{j \leq k, d \in \widehat{\mathcal{D}}, d_{\text{ne}(i)} = \sigma} \alpha_{j,d} u_j \right)
\end{aligned}$$

which lives in the span of the vectors $v^{(i),\sigma} = \sum_{j \leq k, d \in \widehat{\mathcal{D}}, d_{\text{ne}(i)} = \sigma} \alpha_{j,d} u_j$ regardless of c. The number of such vectors is $|\{0,1\}^{|\text{ne}(i)|}| = 2^{|\text{ne}(i)|}$. □

A.3 Linearity from the implicit bias of gradient descent

In this section, we will prove Theorem 4 and Theorem 5 and additional auxiliary theorems.

Theorem 4 (Gradient descent with fixed embeddings)**.** *Fix $i \in [m]$. Let $\widehat{\mathcal{D}} = \{d^{\diamond}_{(i\to 1)}, d^{\diamond}_{(i\to 0)}\}$ where $d^{\diamond} = [\diamond, ..., \diamond]$ and $\Delta_{c,i} = g(c_{(i\to 1)}) - g(c_{(i\to 0)})$. Suppose the loss function is the fol-*

lowing:

$$L(\{\Delta_{c,i}\}_c, f(d^\diamond_{(i\to 1)}), f(d^\diamond_{(i\to 0)})) = \sum_{c\in\overline{\mathcal{C}}} \Big(\exp(-\Delta_{c,i}^T f(d^\diamond_{(i\to 1)})) + \exp(\Delta_{c,i}^T f(d^\diamond_{(i\to 0)})) \Big)$$

where for all $c \in \overline{\mathcal{C}}$, $c_i = 1$ and $f(d^\diamond_{(i\to 1)})) \neq f(d^\diamond_{(i\to 0)})$. Then fixing f and training g using gradient descent with the appropriate step size, we have

$$\lim_{t\to\infty} \cos(\Delta^t_{c^1,i}, \Delta^t_{c^2,i}) = 1$$

for any $c^1, c^2 \in \overline{\mathcal{C}}$ where the superscript t is meant to represent vectors after t number of iterations.

Proof. First note that we can break the whole optimization problem into smaller subproblems where each subproblem only depends on one counterfactual pair. By Theorem 3 of [Soudry et al., 2018], all $\Delta_{c,i}$ converges to have the same direction as the hard margin SVM solution. □

Proposition 31. *Suppose $p(c) > 0$ for any $c \in \widehat{\mathcal{C}} = \mathcal{C}$ and $|\hat{p}(c|d) - p(c|d)| < \varepsilon$ for all $c \in \widehat{\mathcal{C}}$ and $d \in \widehat{\mathcal{D}}$ such that $0 < \varepsilon < p(c|d)$ for all c, d where $p(c|d) > 0$. Then for any latent variable C_i, we have that*

$$\exp(-(g(c_{(i\to 1)}) - g(c_{(i\to 0)}))^T (f(d_{(i\to 0)}) - f(d_{(i\to 1)}))) < \frac{\varepsilon^2}{\big(p(c_{c_i=0}|d_{c_i=0}) - \varepsilon\big)\big(p(c_{c_i=1}|d_{c_i=1}) - \varepsilon\big)}$$

for any $c \in \widehat{\mathcal{C}}$ and any $d \in \widehat{\mathcal{D}}$.

Proof. For any $c \in \widehat{\mathcal{C}}$ and $d \in \widehat{\mathcal{D}}$, we have that

$$\frac{p(c_{c_i=1}|d_{c_i=0})}{p(c_{c_i=0}|d_{c_i=0})} = 0, \qquad \frac{p(c_{c_i=0}|d_{c_i=1})}{p(c_{c_i=1}|d_{c_i=1})} = 0$$

These ratios are well-defined because $p(c) > 0$ for any c. Therefore,

$$\frac{\hat{p}(c_{c_i=1}|d_{c_i=0})}{\hat{p}(c_{c_i=0}|d_{c_i=0})} < \frac{\varepsilon}{p(c_{c_i=0}|d_{c_i=0}) - \varepsilon}, \qquad \frac{\hat{p}(c_{c_i=0}|d_{c_i=1})}{\hat{p}(c_{c_i=1}|d_{c_i=1})} < \frac{\varepsilon}{p(c_{c_i=1}|d_{c_i=1}) - \varepsilon}$$

Thus,

$$\begin{aligned} &\exp(-(g(c_{(c_i=1)}) - g(c_{(c_i=0)}))^T(f(d_{(c_i=0)}) - f(d_{(c_i=1)}))) \\ &\qquad = \frac{\hat{p}(c_{c_i=1}|d_{c_i=0})}{\hat{p}(c_{c_i=0}|d_{c_i=0})}\frac{\hat{p}(c_{c_i=0}|d_{c_i=1})}{\hat{p}(c_{c_i=1}|d_{c_i=1})} < \frac{\varepsilon^2}{\big(p(c_{c_i=0}|d_{c_i=0}) - \varepsilon\big)\big(p(c_{c_i=1}|d_{c_i=1}) - \varepsilon\big)} \end{aligned}$$

□

Theorem 32. *Given loss function $L(u, v) = \exp(-u^T v)$, any starting point u_0, v_0 where $u_0 \neq -\alpha v_0$ for some $\alpha > 0$, and any step size $\eta < \frac{1}{L(u_0,v_0)}$, the gradient descent iterates will have the following properties:*

(a) $\lim_{t\to\infty} L(t) = \lim_{t\to\infty} L(u_t, v_t) = 0$

(b) $\lim_{t\to\infty} ||u_t|| \to \infty$ *and* $\lim_{t\to\infty} ||v_t|| \to \infty$

(c) $\cos(u_t, v_t)$ *increases monotonically with* t

(d) $\lim_{t\to\infty} \cos(u_t, v_t) = 1$

Proof. Before proving the theorem, let's write out a few equations. By the gradient descent algorithm, we have the following equations:

$$u_{t+1} = u_t + \eta L(t) v_t$$

$$v_{t+1} = v_t + \eta L(t) u_t$$

Thus,

$$
\begin{aligned}
||u_{t+1}||^2 &= ||u_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||v_t||^2 \\
||v_{t+1}||^2 &= ||v_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||u_t||^2 \\
\langle u_{t+1}, v_{t+1}\rangle &= (1 + \eta^2 L(t)^2)\langle u_t, v_t\rangle + \eta L(t)(||u_t||^2 + ||v_t||^2)
\end{aligned}
$$

Now let's prove each claim one by one.

First of all, we know that

$$\langle u_{t+1}, v_{t+1}\rangle - \langle u_t, v_t\rangle = \eta^2 L(t)^2 \langle u_t, v_t\rangle + \eta L(t)(||u_t||^2 + ||v_t||^2)$$

The difference is positive if $\langle u_t, v_t\rangle > 0$. To deal with the case of a negative inner product, we will use induction to prove that for any t, $\langle u_{t+1}, v_{t+1}\rangle - \langle u_t, v_t\rangle$ is positive and thus $L(t)$ decreases monotonically.

Base case ($t = 0$): The difference is positive if $\langle u_0, v_0\rangle > 0$. Let's consider the case when $\langle u_0, v_0\rangle \leq 0$

$$
\begin{aligned}
\langle u_1, v_1\rangle - \langle u_0, v_0\rangle &= \eta^2 L(0)^2 \langle u_0, v_0\rangle + \eta L(0)(||u_0||^2 + ||v_0||^2) \\
&= (\eta^2 L(0)^2 - 2\eta L(0))\langle u_0, v_0\rangle + \eta L(0)(u_0 + v_0)^2 \\
&= \eta L(0)(\eta L(0) - 2)\langle u_0, v_0\rangle + \eta L(0)(u_0 + v_0)^2 > 0
\end{aligned}
$$

The last inequality is due to $\eta < \frac{1}{L(u_0, v_0)}$.

Inductive step : Again, the difference is positive if $\langle u_t, v_t\rangle > 0$. Let's consider the case when $\langle u_t, v_t\rangle \leq 0$

$$
\begin{aligned}
\langle u_{t+1}, v_{t+1}\rangle - \langle u_t, v_t\rangle &= \eta^2 L(t)^2 \langle u_t, v_t\rangle + \eta L(t)(||u_t||^2 + ||v_t||^2) \\
&= (\eta^2 L(t)^2 - 2\eta L(t))\langle u_t, v_t\rangle + \eta L(t)(u_t + v_t)^2 \\
&= \eta L(t)(\eta L(t) - 2)\langle u_t, v_t\rangle + \eta L(t)(u_t + v_t)^2 > 0
\end{aligned}
$$

The last inequality is due to $\eta < \frac{1}{L(u_0,v_0)}$ and the inductive hypothesis that $L(t)$ decreases monotonically.

It is worth noting because $\langle u_t, v_t \rangle$ is monotonically increasing, there must exist a time t_p such that $\langle u_{t_p}, v_{t_p} \rangle > 0$. If the opposite is true, then $\langle u_t, v_t \rangle$ must converge to a nonpositive number which is not possible because the difference between consecutive numbers in the sequence is strictly positive unless both u_t and v_t converges to zero. In other words, if $\langle u_t, v_t \rangle$ does not diverge, it must be a Cauchy sequence which needs both u_t and v_t to reach 0. Suppose $u_t \to 0$ and $v_t \to 0$. We know that

$$u_{t+1} + v_{t+1} = (1 + \eta L(t))(u_t + v_t) \tag{A.3.1}$$

which means the only possible scenario that $u_t \to 0$ and $v_t \to 0$ is when $u_0 + v_0 = 0$ which is excluded by the assumption on initial u_0 and v_0.

Because $L(t) > 0$, we know that $\lim_{t\to\infty} L(t)$ has a limit. Suppose the limit is some constant $c_1 \neq 0$. Then, we have

$$\begin{aligned}
\langle u_T, v_T \rangle - \langle u_{t_p}, v_{t_p} \rangle &= \sum_{t=t_p}^{T-1} \big(\langle u_{t+1}, v_{t+1} \rangle - \langle u_t, v_t \rangle \big) \\
&= \sum_{t=t_p}^{T-1} \eta^2 L(t)^2 \langle u_t, v_t \rangle + \sum_{t=t_p}^{T-1} \eta L(t) (||u_t||^2 + ||v_t||^2) \\
&\geq \sum_{t=t_p}^{T-1} \eta^2 c_1^2 \langle u_{t_p}, v_{t_p} \rangle \\
&> \sum_{t=t_p}^{T-1} C
\end{aligned}$$

for some constant C. This would imply $\langle u_T, v_T \rangle \to \infty$ which contradicts that $\lim_{t\to\infty} L(t) > 0$. Therefore, $\lim_{t\to\infty} L(t) = 0$.

For the second property, we already know at least one of $||u_t||$ and $||v_t||$ will converge to infinity for the loss to converge to zero. Suppose one of them converges to a constant. Without loss of generality, let's assume $\lim_{t\to\infty} ||u_t|| \to \infty$ and for all t, $||v_t|| \leq C_v$ for some constant C_v. This implies that $\lim_{t\to\infty} \frac{||v_t||}{||u_t||} \to 0$. On the other hand, let's consider the following equation of $q_t = \frac{||v_t||}{||u_t||}$ for $t \geq t_p$:

$$q_{t+1}^2 = \frac{||v_{t+1}||^2}{||u_{t+1}||^2} = \frac{||v_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||u_t||^2}{||u_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||v_t||^2}$$

If $q_t^2 < 1$, then

$$\frac{2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||u_t||^2}{2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||v_t||^2} > 1$$

Therefore, if $q_t^2 < 1$, then $q_{t+1}^2 > q_t^2$. Similarly, if $q_t^2 > 1$, then $q_{t+1}^2 < q_t^2$. As a result, q_t will not converge to zero, which is a contradiction.

For the third property, let's consider this equation.

$$\begin{aligned}\cos(u_{t+1}, v_{t+1})^2 &= \frac{(\langle u_{t+1}, v_{t+1}\rangle)^2}{||u_{t+1}||^2 ||v_{t+1}||^2} \\ &= \frac{(\langle u_{t+1}, v_{t+1}\rangle)^2}{(||u_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||v_t||^2)(||v_t||^2 + 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2 ||u_t||^2)}\end{aligned}$$

where

$$\begin{aligned}(\langle u_{t+1}, v_{t+1}\rangle)^2 = (\langle u_t, v_t\rangle)^2 + (2\eta^2 L(t)^2 + \eta^4 L(t)^4)(\langle u_t, v_t\rangle)^2 + \eta^2 L(t)^2 (||u_t||^4 + ||v_t||^4) \\ + 2(1 + \eta^2 L(t)^2)(\eta L(t))\langle u_t, v_t\rangle(||u_t||^2 + ||v_t||^2) + 2\eta^2 L(t)^2 ||u_t||^2 ||v_t||^2\end{aligned}$$

Therefore,

$$\cos(u_{t+1}, v_{t+1})^2 = \frac{(\langle u_t, v_t\rangle)^2 + \Delta X}{(||u_t||^2 + \Delta Y)(||v_t||^2 + \Delta Z)}$$

where

$$\begin{aligned}
\Delta X &= (2\eta^2 L(t)^2 + \eta^4 L(t)^4)(\langle u_t, v_t\rangle)^2 + \eta^2 L(t)^2(||u_t||^4 + ||v_t||^4) \\
&\quad + 2(1+\eta^2 L(t)^2)(\eta L(t))\langle u_t, v_t\rangle(||u_t||^2 + ||v_t||^2) + 2\eta^2 L(t)^2||u_t||^2||v_t||^2 \\
\Delta Y &= 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2||v_t||^2 \\
\Delta Z &= 2\eta L(t)\langle u_t, v_t\rangle + \eta^2 L(t)^2||u_t||^2
\end{aligned}$$

then,

$$\begin{aligned}
&\Delta X - (\Delta Y)||v_t||^2 - (\Delta Z)||u_t||^2 - (\Delta Y)(\Delta Z) \\
&\qquad = (\eta^4 L(t)^4 - 2\eta^2 L(t)^2)((\langle u_t, v_t\rangle)^2 - ||u_t||^2||v_t||^2) \\
&\qquad = (2\eta^2 L(t)^2 - \eta^4 L(t)^4)||u_t||^2||v_t||^2(1 - \cos(u_t, v_t)^2)
\end{aligned}$$

This is zero if and only if $\cos(u_t, v_t)^2 = 1$. Otherwise, it is strictly positive. Therefore, if $t \geq t_p$, by Lemma 33, $\cos(u_t, v_t)$ increases monotonically.

Before studying the case of $t < t_p$, let's first consider this difference.

$$\begin{aligned}
\Delta_t &= ||u_t||^2||v_t||^2 - (\langle u_v, v_t\rangle)^2 \\
\Delta_{t+1} - \Delta_t &= (\eta^4 L(t)^4 - 2\eta^2 L(t)^2)||u_t||^2||v_t||^2(1 - \cos(u_t, v_t)^2) < 0
\end{aligned}$$

Therefore, when $t < t_p$, because $\langle u_v, v_t\rangle$ increases but $\langle u_v, v_t\rangle < 0$, $(\langle u_v, v_t\rangle)^2$ decreases. Since Δ_t decreases, $||u_t||^2||v_t||^2$ need to decrease as well. Therefore, $cos(u_t, v_t)$ increases when $t < t_p$.

In fact, because $\Delta_t \geq 0$, Δ_t must have a limit. In particular, this would also imply that Δ_t has an upper bound.

Finally, we can prove the last property. Because $\cos(u_t, v_t)$ increases monotonically and $\cos(u_t, v_t) \leq 1$, it must have a limit. Note that the limit does not have to be 1 for the loss to converge to 0. The key observation is that by (A.3.1), the direction of $u_t + v_t$ is always the same. In fact, it is reasonable to guess that both u_t and v_t will converge in that direction. Let's project u_{t+1} onto the orthogonal complement of $\text{span}(u_t + v_t)$. And we want to show that the projected vector is bounded. Therefore, we only need to consider $t > t_p$.

$$||u_{t+1}^{\perp}||^2 = ||u_{t+1}||^2 - ||\frac{\langle u_{t+1}, u_t + v_t \rangle}{||u_t + v_t||}||^2$$

then,

$$||u_{t+1}^{\perp}||^2 = \frac{(1-\eta L(t))^2(||u_t||^2||v_t||^2 - (\langle u_v, v_t \rangle)^2)}{||u_t + v_t||^2} \leq \frac{(1-\eta L(t))^2(||u_t||^2||v_t||^2 - (\langle u_v, v_t \rangle)^2)}{||u_{t_p} + v_{t_p}||^2}$$

We already know that the numerator is bounded. Therefore, $||u_{t+1}^{\perp}||$ is also bounded. Similarly, $||v_{t+1}^{\perp}||$ is also bounded.

We know that both $||u_t||$ and $||v_t||$ diverge to infinity, thus $\lim_{t\to\infty} \cos(u_t, v_t + u_t) = 1$ and $\lim_{t\to\infty} \cos(v_t, v_t + u_t) = 1$. Because of the following well-known inequality,

$$\begin{aligned} \cos(u_t, v_t + u_t)\cos(v_t, v_t + u_t) - \sqrt{1 - \cos(u_t, v_t + u_t)^2}\sqrt{1 - \cos(v_t, v_t + u_t)^2} \\ \leq \cos(u_t, v_t) \\ \leq \cos(u_t, v_t + u_t)\cos(v_t, v_t + u_t) + \sqrt{1 - \cos(u_t, v_t + u_t)^2}\sqrt{1 - \cos(v_t, v_t + u_t)^2} \end{aligned}$$

we have $\lim_{t\to\infty} \cos(u_t, v_t) = 1$ □

Lemma 33. *Let $X, \Delta X, Y, \Delta Y, Z, \Delta Z > 0$ where $YZ \neq 0$ and $(Y + \Delta Y)(Z + \Delta Z) \neq 0$, if*

$$\Delta X > (\Delta Y)Z + (\Delta Z)Y + (\Delta Y)(\Delta Z), \quad \frac{X}{YZ} < 1$$

then,

$$\frac{X + \Delta X}{(Y + \Delta Y)(Z + \Delta Z)} > \frac{X}{YZ}$$

Proof. The proof is straightforward. □

Finally, we will prove our main Theorem 5, which is equivalent to the following theorem with simplified notation.

Theorem 34. *Given loss function*

$$L(u^{(1)}, u^{(2)}, \{v^{(i)}\}_{i=1}^{K}) = \sum_{i=1}^{K} \big(\exp(-\langle u^{(1)}, v^{(i)} \rangle) + \exp(\langle u^{(2)}, v^{(i)} \rangle) \big)$$

Suppose at starting time, $\langle u_0^{(1)}, u_0^{(2)} \rangle = 0$*,* $\langle u_0^{(1)}, v_0^{(i)} \rangle = 0$*,* $\langle u_0^{(2)}, v_0^{(i)} \rangle = 0$ *and* $\langle v_0^{(i)}, v_0^{(j)} \rangle = 0$*,* $||u_0^{(1)}|| = ||u_0^{(2)}|| = C_u$*,* $||v_0^{(i)}|| = C_v$ *for some positive constants* C_v*,* C_u *and all* $i, j \in [K]$*, then the gradient descent iterates will have the following properties:*

(a) $\lim_{t\to\infty} \cos(v_t^{(i)}, v_t^{(j)}) = 1$ *for all* $i, j \in [K]$

(b) $\lim_{t\to\infty} \cos(u_t^{(1)} - u_t^{(2)}, v_t^{(i)}) = 1$ *for all* $i \in [K]$

Proof. Before proving the theorem, let's write out a few equations. For simplicity, let's denote $\ell_t^{1,i} = \exp(-\langle u_t^{(1)}, v_t^{(i)} \rangle)$, $\ell_t^{2,i} = \exp(\langle u_t^{(1)}, v_t^{(i)} \rangle)$, $\Delta_t = u_t^{(1)} - u_t^{(2)}$ and $\Delta_t^i = \ell_t^{1,i} u_t^{(1)} - \ell_t^{2,i} u_t^{(2)}$.

By the gradient descent algorithm with learning rate η,

$$\begin{aligned} v_{t+1}^{(i)} &= v_t^{(i)} + \eta \Delta_t^i \\ u_{t+1}^{(1)} &= u_t^{(1)} + \eta \sum_{j=1}^{K} \ell_t^{1,j} v_t^{(j)} \\ u_{t+1}^{(2)} &= u_t^{(2)} - \eta \sum_{j=1}^{K} \ell_t^{2,j} v_t^{(j)} \end{aligned}$$

Furthermore,

$$\begin{aligned}
\Delta_{t+1} &= u^{(1)}_{t+1} - u^{(2)}_{t+1} = u^{(1)}_{t} - u^{(2)}_{t} + \eta \sum_{j=1}^{K} \ell^{1,j}_t v^{(j)}_t + \eta \sum_{j=1}^{K} \ell^{2,j}_t v^{(j)}_t \\
&= \Delta_t + \eta \sum_{j=1}^{K} (\ell^{1,j}_t + \ell^{2,j}_t) v^{(j)}_t \\
\Delta^i_{t+1} &= \ell^{1,i}_{t+1} u^{(1)}_{t+1} - \ell^{2,i}_{t+1} u^{(2)}_{t+1} = \ell^{1,j}_{t+1} u^{(1)}_{t} + \eta \ell^{1,j}_{t+1} \sum_{j=1}^{K} \ell^{1,j}_t v^{(j)}_t - \ell^{2,i}_{t+1} u^{(2)}_{t} + \eta \ell^{2,j}_{t+1} \sum_{j=1}^{K} \ell^{2,j}_t v^{(j)}_t \\
&= \ell^{1,i}_{t+1} u^{(1)}_{t} - \ell^{2,i}_{t+1} u^{(2)}_{t} + \eta \big(\ell^{1,i}_{t+1} \sum_{j=1}^{K} \ell^{1,j}_t v^{(j)}_t + \ell^{2,i}_{t+1} \sum_{j=1}^{K} \ell^{2,j}_t v^{(j)}_t \big)
\end{aligned}$$

With our initial condition, by symmetry, one can show that for any $t \geq 1$.

(a) $\ell^{1,i}_t = \ell^{1,j}_t = \ell^{2,i}_t = \ell^{2,j}_t$ for any $i, j \in [K]$ and the loss decreases monotonically.

(b) $\Delta^i_t = \Delta^j_t$.

(c) $\langle v^{(i)}_t, \sum_{j=1}^{K} v^{(j)}_t \rangle = C^1_t$ for some positive constant C^1_t that does not depend on index i. And C^1_t increases monotonically with t.

(d) $\langle v^{(i)}_t, \Delta^i_t \rangle = C^2_t$ for some positive constant C^2_t that does not depend on index i.

(e) $\langle u^{(1)}_t, \Delta^i_t \rangle = -\langle u^{(2)}_t, \Delta^i_t \rangle = C^3_t$ for some positive constant C^3_t that does not depend on index i.

(f) $\langle u^{(1)}_t, v^i_t \rangle = -\langle u^{(2)}_t, v^i_t \rangle = C^4_t$ for some positive constant C^4_t that does not depend on index i.

We'll prove this by induction.

Base step ($t = 1$) First of all, by the initial condition, $\ell_t^{1,i} = \ell_t^{1,j} = \ell_t^{2,i} = \ell_t^{2,j} = 1$.

$$\begin{aligned}
\ell_1^{1,i} &= \exp(-\langle u_1^{(1)}, v_1^{(i)}\rangle) = \exp\big(-\langle u_0^{(1)} + \eta\sum_{j=1}^{K} \ell_0^{1,i} v_0^{(j)}, v_0^{(i)} + \eta\Delta_0^i\rangle\big) \\
&= \exp\big(-\langle u_0^{(1)}, v_0^{(i)}\rangle - \eta\langle u_0^{(1)}, \Delta_0^i\rangle - \eta\langle v_0^{(i)}, \sum_{j=1}^{K} \ell_0^{1,j} v_0^{(j)}\rangle - \eta^2\langle \Delta_0^i, \sum_{j=1}^{K} \ell_0^{1,i} v_0^{(i)}\rangle\big) \\
&= \exp\big(-\eta\langle u_0^{(1)}, \Delta_0^i\rangle - \eta\ell_0^{1,i}||v_0^{(i)}||^2\big) = \exp\big(-\eta\ell_0^{1,i}||u_0^{(1)}||^2 - \eta\ell_0^{1,i}||v_0^{(i)}||^2\big) \\
&= \exp\big(-\eta\ell_0^{1,i}C_u^2 - \eta\ell_0^{1,i}C_v^2\big) = \exp\big(-\eta C_u^2 - \eta C_v^2\big) \\
&= \ell_1^{1,j}
\end{aligned}$$

Similarly

$$\begin{aligned}
\ell_1^{2,i} &= \exp(\langle u_1^{(2)}, v_1^{(i)}\rangle) = \exp\big(\langle u_0^{(2)} - \eta\sum_{j=1}^{K} \ell_0^{2,i} v_0^{(j)}, v_0^{(i)} + \eta\Delta_0^i\rangle\big) \\
&= \exp\big(\langle u_0^{(2)}, \Delta_0^i\rangle - \eta\ell_0^{1,i}||v_0^{(i)}||^2\big) = \exp\big(-\eta C_u^2 - \eta C_v^2\big) \\
&= \ell_1^{2,j} = \ell_1^{1,i} = \ell_1^{1,j}
\end{aligned}$$

For simplicity, let use $\ell_1 = \ell_1^{1,i} = \ell_1^{2,i}$. This also implies that $\Delta_1^i = \Delta_1^j = \ell_1\Delta_1$.

On the other hand,

$$\begin{aligned}
\langle v_1^{(i)}, \sum_{j=1}^{K} v_1^{(j)}\rangle &= \langle v_0^{(i)} + \eta\Delta_0^i, \sum_{j=1}^{K}(v_0^{(j)} + \eta\Delta_0^i)\rangle \\
&= ||v_0^{(i)}||^2 + \eta^2\langle \Delta_0^i, \sum_{j=1}^{K} \Delta_0^j\rangle \\
&= ||v_0^{(i)}||^2 + \eta^2 K||\Delta_0^i||^2 = C_v^2 + 2\eta^2 K C_u^2 \\
&> \langle v_0^{(i)}, \sum_{j=1}^{K} v_0^{(j)}\rangle = ||v_0^{(i)}||^2 = C_v^2
\end{aligned}$$

Notice that the constant does not depend on i.

$$\begin{aligned}\langle v_1^{(i)}, \Delta_1^i \rangle &= \ell_1 \langle v_0^{(i)} + \eta\Delta_0^i, \Delta_1 \rangle = \ell_1 \langle v_1^{(i)}, \Delta_0 + \eta \sum_{j=1}^{K} (\ell_0^{1,j} + \ell_0^{2,j}) v_0^{(j)} \rangle \\ &= \ell_1 \langle v_0^{(i)} + \eta\Delta_0^i, \Delta_0 + 2\eta \sum_{j=1}^{K} v_0^{(j)} \rangle = 2\ell_1\eta ||v_0^{(i)}||^2 + \ell_1\eta(||u_0^{(1)}||^2 + ||u_0^{(2)}||^2) \\ &= 2\ell_1\eta C_v^2 + 2\eta\ell_1||C_u||^2\end{aligned}$$

Notice that the constant does not depend on i.

$$\begin{aligned}\langle u_1^{(1)}, \Delta_1^i \rangle &= \ell_1 \langle u_1^{(1)}, \Delta_1 \rangle = \ell_1 \langle u_0^{(1)} + \eta \sum_{j=1}^{K} \ell_0^{1,i} v_0^{(j)}, \Delta_0 + \eta \sum_{j=1}^{K} (\ell_0^{1,j} + \ell_0^{2,j}) v_0^{(j)} \rangle \\ &= \ell_1 \langle u_0^{(1)} + \eta \sum_{j=1}^{K} v_0^{(j)}, \Delta_0 + 2\eta \sum_{j=1}^{K} v_0^{(j)} \rangle = \ell_1 ||u_0^{(1)}||^2 + 2\ell_1\eta^2 \sum_{j=1}^{K} ||v_0^{(j)}||^2 \\ &= \ell_1 C_u^2 + 2\ell_1\eta^2 K ||C_v||^2 \\ &= -\langle u_1^{(2)}, \Delta_1^i \rangle\end{aligned}$$

Again, the constant does not depend on i.

Finally,

$$\begin{aligned}\langle u_1^{(1)}, v_1^i \rangle &= \langle u_0^{(1)} + \eta \sum_{i=1}^{K} \ell_0^{1,i} v_0^{(i)}, v_0^{(i)} + \eta\Delta_0^i \rangle \\ &= \langle u_0^{(1)} + \eta \sum_{i=1}^{K} v_0^{(i)}, v_0^{(i)} + \eta\Delta_0^i \rangle \\ &= \eta ||u_0^{(1)}||^2 + \eta ||v_0^{(i)}||^2 = \eta C_u^2 + \eta C_v^2 \\ &= -\langle u_1^{(2)}, v_1^i \rangle\end{aligned}$$

Inductive step By the inductive hypothesis, let $\ell_t = \ell_t^{1,i} = \ell_t^{2,i}$.

$$\begin{aligned}
\ell^{1,i}_{t+1} &= \exp(-\langle u^{(1)}_{t+1}, v^{(i)}_{t+1}\rangle) = \exp\big(-\langle u^{(1)}_t + \eta\sum_{j=1}^{K}\ell^{1,i}_t v^{(j)}_t, v^{(i)}_t + \eta\Delta^i_t\rangle\big) \\
&= \exp\big(-\langle u^{(1)}_t, v^{(i)}_t\rangle - \eta\langle u^{(1)}_t, \Delta^i_t\rangle - \eta\langle v^{(i)}_t, \sum_{j=1}^{K}\ell^{1,j}_t v^{(j)}_t\rangle - \eta^2\langle\Delta^i_t, \sum_{j=1}^{K}\ell^{1,i}_t v^{(i)}_t\rangle\big) \\
&= \exp\big(-C^4_t - \eta C^3_t - \eta\ell_t C^1_t - \eta^2\ell_t K C^2_t\big) \\
&= \ell^{1,j}_{t+1} = \ell^{2,i}_{t+1} = \ell^{2,j}_{t+1}
\end{aligned}$$

$$\begin{aligned}
\langle v^{(i)}_{t+1}, \sum_{j=1}^{K} v^{(j)}_{t+1}\rangle &= \langle v^{(i)}_t + \eta\Delta^i_t, \sum_{j=1}^{K}(v^{(j)}_t + \eta\Delta^i_t)\rangle \\
&= \langle v^{(i)}_t, \sum_{j=1}^{K} v^{(j)}_t\rangle + \eta\langle v^{(i)}_t, \sum_{j=1}^{K}\Delta^i_t\rangle + \eta\langle\Delta^i_t, \sum_{j=1}^{K} v^{(j)}_t\rangle + \eta^2\langle\Delta^i_t, \sum_{j=1}^{K}\Delta^i_t\rangle \\
&= C^1_t + 2\eta K C^t_2 + \eta^2\ell_t\langle u^{(1)}_t - u^{(2)}_t, \sum_{j=1}^{K}\Delta^i_t\rangle \\
&= C^1_t + 2\eta K C^t_2 + \eta^2\ell_t\langle u^{(1)}_t - u^{(2)}_t, \sum_{j=1}^{K}\Delta^i_t\rangle \\
&= C^1_t + 2\eta K C^t_2 + 2\eta^2\ell_t K C^4_t > \langle v^{(i)}_t, \sum_{j=1}^{K} v^{(j)}_t\rangle > 0
\end{aligned}$$

By same logic, one show the inductive step for $\langle v^{(i)}_t, \Delta^i_t\rangle$, $\langle u^{(1)}_t, \Delta^i_t\rangle$, $\langle u^{(2)}_t, \Delta^i_t\rangle$, $\langle u^{(1)}_t, v^i_t\rangle$ and $\langle u^{(2)}_t, v^i_t\rangle$.

Therefore, we can simplify the notation. The gradient descent iterates can be rewritten as:

$$v_{t+1}^{(i)} = v_t^{(i)} + \eta\ell_t\Delta_t \quad u_{t+1}^{(1)} = u_t^{(1)} + \eta\ell_t\sum_{i=1}^{K} v_t^{(i)}$$

$$u_{t+1}^{(2)} = u_t^{(2)} - \eta\ell_t\sum_{i=1}^{K} v_t^{(i)} \quad \Delta_{t+1} = \Delta_t + 2\eta\ell_t\sum_{j=1}^{K} v_t^{(j)}$$

Furthermore, let $V_t = \sum_{j=1}^{K} v_t^{(j)}$

$$V_{t+1} = V_t + K\eta\ell_t\Delta_t$$

$$\Delta_{t+1} = \Delta_t + 2\eta\ell_t V_t$$

In essence, the rest of the proof follows from the proof of Theorem 32 and symmetry.

Loss converging to zero One first notice that by the induction argument above, the loss must be monotonically decreasing. In fact, $\lim_{t\to\infty}\ell_t = 0$. To see this, notice that

$$\langle u_{t+1}^{(1)}, v_{t+1}^{(i)}\rangle - \langle u_t^{(1)}, v_t^{(i)}\rangle \geq \eta\langle v_t^{(j)}, \ell_t\sum_{j=1}^{K} v_t^{(j)}\rangle \geq \ell_t\eta\langle v_1^{(j)}, \sum_{j=1}^{K} v_t^{(j)}\rangle = \ell_t\eta C_1^1$$

Suppose ℓ_t is not converging to zero. Then ℓ_t has an lower bound, which means $\langle u_{t+1}^{(1)}, v_{t+1}^{(i)}\rangle$ will increase to infinity. This is a contradiction. Therefore, $\lim_{t\to\infty}\ell_t = 0$.

In fact, one can also show that, $\lim_{t\to\infty}||V_t|| \to \infty$ and $\lim_{t\to\infty}||\Delta_t|| \to \infty$. To see this, one first notice that,

$$\lim_{t\to\infty}\exp(-\langle\Delta_t, V_t\rangle) = \lim_{t\to\infty}\ell_t^{2K} \to 0$$

Therefore, at least one of $||V_t||$ or $||\Delta_t||$ needs to go to infinity. Suppose only $||\Delta_t||$ reaches infinity, then $\lim_{t\to\infty}\frac{||V_t||}{||\Delta_t||} \to 0$. On the other hand,

$$\frac{||V_{t+1}||^2}{||\Delta_{t+1}||^2} = \frac{||V_t||^2 + 2K\eta\ell_t\langle V_t, \Delta_t\rangle + K^2\eta^2\ell^2||\Delta_t||^2}{||\Delta_t||^2 + 4\eta\ell_t\langle V_t, \Delta_t\rangle + 4\eta^2\ell^2||V_t||^2}$$

Suppose $\frac{||V_t||^2}{||\Delta_t||^2} < K/2$, then

$$\frac{2K\eta\ell_t\langle V_t, \Delta_t\rangle + K^2\eta^2\ell^2||\Delta_t||^2}{4\eta\ell_t\langle V_t, \Delta_t\rangle + 4\eta^2\ell^2||V_t||^2} \geq K/2$$

Therefore, if $\frac{||V_t||^2}{||\Delta_t||^2} < K/2$, $\frac{||V_{t+1}||^2}{||\Delta_{t+1}||^2} > \frac{||V_t||^2}{||\Delta_t||^2}$. Similarly, if $\frac{||V_t||^2}{||\Delta_t||^2} > K/2$, $\frac{||V_{t+1}||^2}{||\Delta_{t+1}||^2} < \frac{||V_t||^2}{||\Delta_t||^2}$. Therefore, $\lim_{t\to\infty} \frac{||V_t||}{||\Delta_t||}$ will not be zero. So both $\lim_{t\to\infty} ||V_t|| \to \infty$ and $\lim_{t\to\infty} ||\Delta_t|| \to \infty$.

Cosine similarity between $v_t^{(i)}, v_t^{(j)}$ converges to one Note for any i, $\lim_{t\to\infty} ||v_t^{(i)}|| \to \infty$. This is because by symmetry,

$$\lim_{t\to\infty} ||v_t^{(i)}|| \geq \lim_{t\to\infty} \frac{||V_t||}{K} \to \infty$$

Then,

$$v_T^{(i)} = v_0^{(i)} + \eta \sum_{t=0}^{T-1} \Delta_t^w$$

Let's denote $D_T = \eta \sum_{t=0}^{T-1} \Delta_t^w$. Then $||v_0^{(i)}|| + ||D_T|| \geq ||v_T^{(i)}||$. Thus, $\lim_{T\to\infty} ||D_T|| = \infty$.

Finally

$$\begin{aligned}\cos(v_t^{(i)}, v_t^{(i)}) &= \frac{\langle v_0^{(i)} + D_T, v_0^{(j)} + D_T\rangle}{||v_t^{(i)}||||v_t^{(j)}||} \geq \frac{\langle v_0^{(i)} + D_T, v_0^{(j)} + D_T\rangle}{(||v_0^{(i)}|| + ||D_t||)(||v_0^{(j)}|| + ||D_t||)} \\ &= \frac{\langle v_0^{(i)} + v_0^{(j)}, D_T\rangle + ||D_T||^2}{(||v_0^{(i)}|| + ||D_t||)(||v_0^{(j)}|| + ||D_t||)} = \frac{\langle v_0^{(i)} + v_0^{(j)}, D_T\rangle/||D_t||^2 + 1}{(||v_0^{(i)}||/||D_t|| + 1)(||v_0^{(j)}||/||D_t|| + 1)}\end{aligned}$$

Thus,

$$\lim_{t\to\infty} \cos(v_t^{(i)}, v_t^{(i)}) = 1$$

Cosine similarity between $v_t^{(i)}, \Delta_t$ converges to one Finally, one can show that $\lim_{t\to\infty} \cos(\Delta_t, V_t) \to 1$ follows the same proof of Theorem 32. For completeness, we will present the full proof here.

Let's first do a simple variable change,

$$\begin{aligned}\sqrt{2}V_{t+1} &= \sqrt{2}V_t + \sqrt{2K}\eta\ell_t\sqrt{K}\Delta_t \\ \sqrt{K}\Delta_{t+1} &= \sqrt{K}\Delta_t + \sqrt{2K}\eta\ell_t\sqrt{2}V_t\end{aligned}$$

Let $\tilde{V}_t = \sqrt{2}V_t$, $\tilde{\Delta}_t = \sqrt{K}\Delta_t$, and $\tilde{\eta} = \sqrt{2K}\eta$, then

$$\begin{aligned}\tilde{V}_{t+1} &= \tilde{V}_t + \tilde{\eta}\ell_t\tilde{\Delta}_t \\ \tilde{\Delta}_{t+1} &= \tilde{\Delta}_t + \tilde{\eta}\ell_t\tilde{V}_t\end{aligned}$$

One first notice that, $\tilde{V}_t + \tilde{\Delta}_t$ always has the direction at any t. Therefore, let's consider the $\tilde{V}_{t+1}^{\perp}$ which is the residual after projecting onto the direction of $\tilde{V}_t + \tilde{\Delta}_t$,

$$\begin{aligned}||\tilde{V}_{t+1}^{\perp}||^2 &= ||\tilde{V}_{t+1}||^2 - ||\frac{\langle \tilde{V}_{t+1}, \tilde{V}_t + \tilde{\Delta}_t\rangle}{||\tilde{V}_t + \tilde{\Delta}_t||}||^2 \\ &= \frac{(1-\tilde{\eta}\ell_t)^2\big(||\tilde{V}_t||^2||\tilde{\Delta}_t||^2 - (\langle \tilde{V}_t, \tilde{\Delta}_t\rangle)^2\big)}{||\tilde{V}_t + \tilde{\Delta}_t||^2} \\ &\leq C_\eta \frac{||\tilde{V}_t||^2||\tilde{\Delta}_t||^2 - (\langle \tilde{V}_t, \tilde{\Delta}_t\rangle)^2}{||\tilde{V}_t + \tilde{\Delta}_t||^2}\end{aligned}$$

Note that $\tilde{\eta}\ell_t$ converges to zero. Therefore, there's an upper bound C_η on $(1-\tilde{\eta}\ell_t)^2$.

On the other hand, let $O_t = ||\tilde{V}_t||^2||\tilde{\Delta}_t||^2 - (\langle \tilde{V}_t, \tilde{\Delta}_t\rangle)^2$. Then

$$O_{t+1} - O_t = (\tilde{\eta}^4\ell_t^4 - 2\tilde{\eta}^2\ell_t^2)||\tilde{V}_t||^2||\tilde{\Delta}_t||^2(1 - \cos(\tilde{V}_t, \tilde{\Delta}_t))$$

Once again, because ℓ_t is eventually converging to zero, O_t will decrease at some point. This is because $\tilde{\eta}^4\ell_t^4 - 2\tilde{\eta}^2\ell_t^2 < 0$ if $\ell_t < \frac{\sqrt{2}}{\tilde{\eta}}$ and $||\tilde{V}_t||^2||\tilde{\Delta}_t||^2(1-\cos(\tilde{V}_t, \tilde{\Delta}_t)) \geq 0$. Because $O_t \geq 0$, it will reach a limit. Therefore, O_t must have an upper bound. Finally, the denominator is diverging and by our inductive statements, it must have a nonzero lower bound.

Therefore $||\tilde{V}_{t+1}^{\perp}||$ is bounded. And as $\lim_{t\to\infty} ||V_t|| \to \infty$ and $\lim_{t\to\infty} ||\Delta_t|| \to \infty$, we have

$$\lim_{t\to\infty} \cos(\tilde{V}_t, \tilde{V}_t + \tilde{\Delta}_t) = 1$$

$$\lim_{t\to\infty} \cos(\tilde{\Delta}_t, \tilde{V}_t + \tilde{\Delta}_t) = 1$$

Thus,

$$\lim_{t\to\infty} \cos(\tilde{\Delta}_t, \tilde{V}_t) = 1$$

This would also imply that $\lim_{t\to\infty} \cos(\Delta_t, v_t^{(i)}) \to 1$ for all i.

□

A.4 Orthogonality

In this section, we will prove our main theorems on orthogonality.

Theorem 7. *Let $\widehat{\mathcal{C}} = \mathcal{C}$ and $\widehat{\mathcal{D}} = \mathcal{D}$. Assuming $p(c) > 0$ for any $c \in \mathcal{C}$ and C_i and C_j are two latent variables separated in G_C. Given any binary vector $c \in \mathcal{C}$, there exists a subset $\mathcal{D}_c \subset \mathcal{D}$ such that $d_i = \diamond$ and $p(c|d) > 0$ for any $d \in \mathcal{D}_c$. If one further assume that $\hat{p}(c|d) = p(c|d)$ for any $d \in \mathcal{D}_c$, then*

$$g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \perp f(d_{(j\to c_j)}) - f(d_{(j\to\diamond)})$$

for any $d \in \mathcal{D}_c$.

Proof. First of all, for any binary vector c, $\mathcal{D}_c$ is non-empty by the positivity assumption because $d^{\diamond} \in \mathcal{D}_c$ where $d^{\diamond} = \{\diamond, ..., \diamond\}$.

Consider an arbitrary $c \in \mathcal{C}$ and an arbitrary $d \in \mathcal{D}_c$. Without loss of generality, let $c_i = 1$. Suppose $d_j = \diamond$, then it must be that $d_{(j\to c_j)} \in \mathcal{D}_c$ because $d_{(j\to c_j)}$ agrees with c on the j-th entry. Similarly, if $d_j = c_j$, then $d_{(j \to \diamond)} \in \mathcal{D}_c$.

By the positivity assumption and the fact that $d_i = \diamond$,

$$\begin{aligned} p(c_{(i\to 1)} | d_{(j\to\diamond)}) > 0, \qquad & p(c_{(i\to 0)} | d_{(j\to\diamond)}) > 0 \\ p(c_{(i\to 1)} | d_{(j\to c_j)}) > 0, \qquad & p(c_{(i\to 0)} | d_{(j\to c_j)}) > 0 \end{aligned}$$

Thus,

$$\begin{aligned} \hat{p}(c_{(i\to 1)} | d_{(j\to\diamond)}) = p(c_{(i\to 1)} | d_{(j\to\diamond)}) \quad & \hat{p}(c_{(i\to 0)} | d_{(j\to\diamond)}) = p(c_{(i\to 0)} | d_{(j\to\diamond)}) \\ \hat{p}(c_{(i\to 1)} | d_{(j\to c_j)}) = p(c_{(i\to 1)} | d_{(j\to c_j)}) \quad & \hat{p}(c_{(i\to 0)} | d_{(j\to c_j)}) = p(c_{(i\to 0)} | d_{(j\to c_j)}) \end{aligned}$$

Then by the Hammersley–Clifford theorem, we can factorize the joint distribution over cliques:

$$p(c) \propto \prod_k \Psi_k(c_{I_k})$$

where $\Psi_k(c_{I_k})$ is a function that only depends on the clique of random variables C_{I_k}.

By this factorization, if $p(c_{(i\to 1)}|d) > 0$ and $p(c_{(i\to 0)}|d) > 0$,

$$\ln \frac{p(c_{(i\to 1)}|d)}{p(c_{(i\to 0)}|d)} = \beta(c_{I_i}, d_{I_i})$$

where β is some function that only depends on cliques that involve C_i. In other words, $i \in I_i$ and $i' \in I_i$ if $C_{i'}$ and C_i are in the same clique in G_C.

Thus,

$$\ln \frac{p(c_{(i\to 1)}|d_{(j\to\diamond)})}{p(c_{(i\to 0)}|d_{(j\to\diamond)})} = \ln \frac{p(c_{(i\to 1)}|d_{(j\to c_j)})}{p(c_{(i\to 0)}|d_{(j\to c_j)})}$$

and,

$$\ln \frac{\hat{p}(c_{(i\to 1)}|d_{(j\to\diamond)})}{\hat{p}(c_{(i\to 0)}|d_{(j\to\diamond)})} = \ln \frac{\hat{p}(c_{(i\to 1)}|d_{(j\to c_j)})}{\hat{p}(c_{(i\to 0)}|d_{(j\to c_j)})}$$

Therefore,

$$\Big(g(c_{(i\to 1)}) - g(c_{(i\to 0)})\Big)^T f(d_{(j\to\diamond)}) = \Big(g(c_{(i\to 1)}) - g(c_{(i\to 0)})\Big)^T f(d_{(j\to c_j)})$$
$$\Big(g(c_{(i\to 1)}) - g(c_{(i\to 0)})\Big)^T \Big(f(d_{(j\to\diamond)}) - f(d_{(j\to c_j)})\Big) = 0$$

□

Corollary 8. *Under the conditions of Theorem 7, if concept C_i and C_j have matched-representations, then*

$$g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \perp g(c_{(j\to 1)}) - g(c_{(j\to 0)})$$

for any $c \in \mathcal{C}$.

Proof. Consider two binary vectors $c^{(0)}, c^{(1)} \in \mathcal{C}$ where $c^{(0)}_j = 0$ and $c^{(1)}_j = 1$ but they agree on other entries.

By Theorem 7,

$$g(c^{(0)}_{(i\to 1)}) - g(c^{(0)}_{(i\to 0)}) \perp f(d^{(0)}_{(j\to c_j)}) - f(d^{(0)}_{(j\to\diamond)})$$

for any $d^{(0)} \in \mathcal{D}_{c^{(0)}}$. Similar statements can be made for $c^{(1)}$ as well.

Note that $\mathcal{D}_{c^{(0)}} \cap \mathcal{D}_{c^{(1)}} \neq \emptyset$ by the positivity assumption. Let $d \in \mathcal{D}_{c^{(0)}} \cap \mathcal{D}_{c^{(1)}}$. Then,

$$g(c^{(0)}_{(i\to 1)}) - g(c^{(0)}_{(i\to 0)}) \perp f(d_{(j\to 0)}) - f(d_{(j\to\diamond)})$$
$$g(c^{(1)}_{(i\to 1)}) - g(c^{(1)}_{(i\to 0)}) \perp f(d_{(j\to 1)}) - f(d_{(j\to\diamond)})$$

By assumption, there exists a unit vector u_i, such that

$$g(c^{(0)}_{(i\to 1)}) - g(c^{(0)}_{(i\to 0)}) = \alpha^{(0)} u_i \quad g(c^{(1)}_{(i\to 1)}) - g(c^{(1)}_{(i\to 0)}) = \alpha^{(1)} u_i$$

for some $\alpha^{(0)}, \alpha^{(1)} > 0$. Therefore,

$$\left\langle u_i, f(d_{(j\to 0)}) - f(d_{(j\to\diamond)}) \right\rangle = \left\langle u_i, f(d_{(j\to 1)}) - f(d_{(j\to\diamond)}) \right\rangle = 0$$

Thus,

$$\left\langle u_i, f(d_{(j\to 0)}) - f(d_{(j\to 1)}) \right\rangle = 0$$

Because latent variable C_i has linaer and matched representations,

$$g(c_{(i\to 1)}) - g(c_{(i\to 0)}) \perp g(c_{(j\to 1)}) - g(c_{(j\to 0)})$$

for any $c \in \mathcal{C}$. □

A.5 Simulated experiments

In this section, we will provide additional details on our simulated experiments with the latent conditional model.

Additional details To let models learn conditional distributions, we train them to make predictions using cross-entropy loss. To turn the aforementioned generated binary vectors into a prediction task, we randomly generate binary masks μ for each vector in the batch. And if $\mu_i = 1$, the i-th entry of the vector is left untouched, and if $\mu_i = 0$, then it is set to -1 which is a numerical representation of the token $\diamond$. The model is trained to use masked vectors to predict the original vectors.

These sampled binary vectors and ternary vectors are mapped into one-hot encodings to avoid neural networks exploiting the inherent structures of these vectors. f and g are modeled as a linear function, which are essentially lookup tables. This construction is made without loss of generality.

Adam optimizer Although the theory is presented with gradient descent, the empirical result is actually robust to the choice of optimizers. We repeat the previous experiments on the complete set of conditionals with Adam optimizer [Kingma and Ba, 2015] using a learning rate 0.001 and observe similar linear representation patterns in Table A.1.

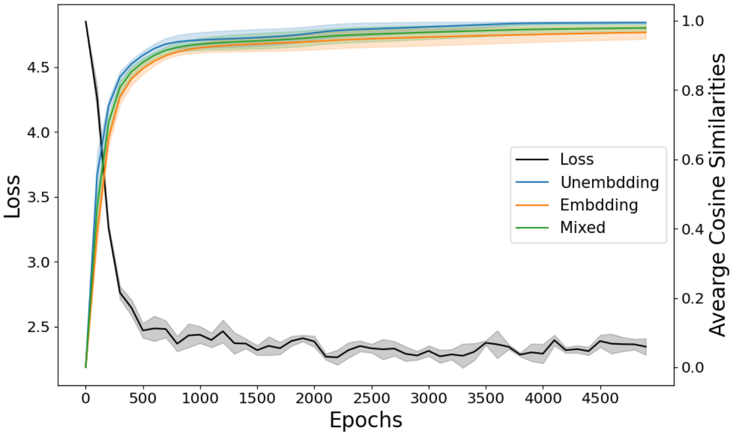


Figure A.1: Loss and cosine similarities change as training progresses. The experiments are tested with 7 hidden variables over 10 runs.

Table A.1: When the model is trained with Adam optimizer, the m latent variables are represented linearly, and the embedding and unembedding representations are matched. The table shows average cosine similarities among and between steering vectors of unembeddings and embeddings. Standard errors are over 100 runs for 3 variables and 4 variables, 50 runs for 5 variables, 20 runs for 6 variables and 10 runs for 7 variables.

m	Unembedding	Embedding	Unembedding and Embedding
3	0.910±0.015	0.923±0.013	0.926±0.012
4	0.972±0.005	0.959±0.005	0.965±0.005
5	0.985±0.004	0.970±0.004	0.975±0.004
6	0.996±0.001	0.977±0.002	0.983±0.001
7	0.966±0.012	0.918±0.016	0.929±0.015

Incomplete set of contexts ($\widehat{\mathcal{D}} \subset \mathcal{D}$) Both the size of $\mathcal{C}$ and $\mathcal{D}$ grow exponentially. To model large language models, not every conditional probability is necessarily trained. In fact, one does not need the complete set of contexts to get linearly encoded representations.

To test this, we run experiments on incomplete subsets of contexts by randomly selecting a few masks $\{\mu\}$. Table A.2 shows that even with subsets of contexts, one can still get linearly encoded representations.

Table A.2: When the model is trained to learn subsets of conditionals for 10 latent variables, the latent variables are still represented linearly and matched. The table shows average cosine similarities. Standard errors are over 10 runs.

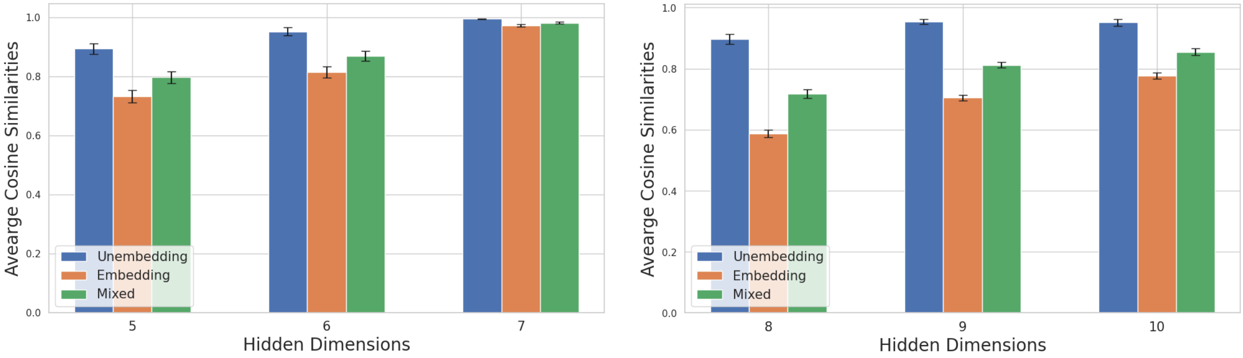
Max Number of Masks	Unembedding	Embedding	Unembedding and Embedding
50	0.974±0.009	0.946±0.014	0.959±0.011
100	0.957±0.009	0.915±0.013	0.934±0.011

Incomplete set of concept vectors or violation of positivity ($\widehat{\mathcal{C}} \subset \mathcal{C}$) In real-world language modeling, not every concept vector will be mapped to a token. We test this by allowing some concept vectors c to have zero probabilities (i.e., $p(c) = 0$). For the experiments, we first randomly select a subset $\hat{\mathcal{C}}$ of $\mathcal{C}$ and then use rejection sampling to collect data points. Table A.3 shows that one can still get reasonable linearity for unembeddings. The embedding alignments drop, however. One possible explanation is due to a lack of training because the size of the problem grows exponentially. On the other hand, Section 2.3.2 suggests the connection between linearity and zero conditional probabilities. Because the positivity assumption is violated, the newly introduced zero conditional probabilities might also cause misalignment.

Change of dimensions Previous experiments set the representation dimension to be the same as the number of latent variables. In this set of experiments, we test how decreasing dimensions affects representations by rerunning experiments on 7 variables with the complete set of contexts and binary vectors as well as experiments with 10 variables with incomplete set of contexts and binary vectors. Figure A.2 shows that although decreasing dimensions

Table A.3: When training with an incomplete set of concept vectors and contexts, unembedding representations are still encoded linearly. The table shows average cosine similarities. Standard errors are over 10 runs.

Number of Hidden Varibles	Unembedding	Embedding	Unembedding and Embedding
10	0.951±0.011	0.777±0.010	0.855±0.011
12	0.896±0.011	0.551±0.008	0.696±0.009



(a) 7 Hidden Variables

(b) 10 Hidden Variables

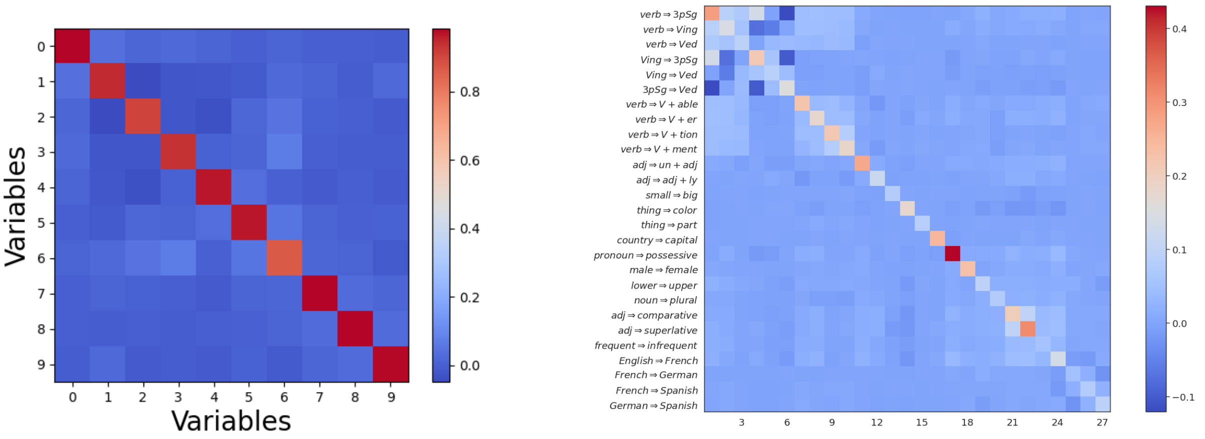
Figure A.2: Average cosine similarities under different hidden dimensions show that reducing dimension dose not hurt linearity significantly.

do make representations less aligned, the effect is not significant.

A.6 Experiments with large language models

In this section, we will provide additional details on our experiments with LLMs.

Examples of counterfactual context and token pairs Example context pairs from OPUS Books [Tiedemann, 2012] that were used to construct the embedding vectors are shown in Table A.4. For the unembedding vectors, we reuse the 27 concepts considered in Park et al. [2023] built atop the work of Gladkova et al. [2016], and they are listed in Table A.5 along with example token pairs.



(a) Simulated Experiments

(b) LLaMA-2

Figure A.3: The pattern of linear and orthogonal representations matches between simulated experiments and LLaMA-2. Specifically, unembedding steering vectors of the same concept in LLaMA-2 have nontrivial alignment, while steering vectors of different concepts are represented almost orthogonally. The heatmap shows the average cosine similarities between different sets of steering vectors for both simulated experiments and LLaMA-2. For simulated experiments, the cosine similarities are averaged over 10 runs.

Winograd Schema We now present full details about our experiments on the Winograd Schema dataset [Levesque et al., 2012]. Recall that we consider context pairs arising from the Winograd Schema, which is a dataset of pairs of sentences that differ in only one or two words and which further contain an ambiguity that can only be resolved with world knowledge and reasoning. Example context pairs from the Winograd Schema [Levesque et al., 2012] are shown in Table A.6. Note the final 2 context pairs in the table, which have ambiguous concepts, thus highlighting the nuances of this dataset as well as of natural language. We filter the dataset to have the ambiguous word near the end of the sentence to better align with our theory.

We again compute the unembedding vectors for these context pairs output by LLaMA-2. For the embedding vectors, since there is no predefined set of concepts to consider (for instance consider the ambiguous examples), we therefore take the difference of the embedding vectors for the first token that differs in these corresponding pairs of sentences. We compute all pairwise cosine similarities. We observe that for non-matching contexts, the average similarity is 0.011 with a maximum of 0.081, whereas for matching contexts, the average similarity is 0.042 with a maximum of 0.161. This aligns with our predictions that the embedding vectors align better with the unembedding vectors of the same concept.

As an additional experiment, we compute similarities between these Winograd context pairs and the 27 concepts from Park et al. [2023] described above. In Table A.7, we display the top 3 pairs of contexts and concepts that have the highest similarities. The alignment seems reasonable as baby, woman, male, and female are different attributes of a person; and wide vs narrow and short vs tall can be construed as different manifestations of small vs big.

Table A.4: Example language context pairs from OPUS Books

Language pair	Context 1	Context 2
French–Spanish	Quinze ou seize, répliqua l'autre	Quince ó diez y seis, replicó el otro
French–German	Comment est-il mon maître?	Wie ist er mein Herr?
English–French	I hesitated for a moment.	J'hésitai une seconde.
German–Spanish	Ich hasse die Spazierfahrten	No me gusta salir en coche.

Table A.5: Concepts and example token pairs, taken from Park et al. [2023]

Concept	Example token pair	Concept	Example token pair
verb ⇒ 3pSg	(accept, accepts)	verb ⇒ Ving	(add, adding)
verb ⇒ Ved	(accept, accepted)	Ving ⇒ 3pSg	(adding, adds)
Ving ⇒ Ved	(adding, added)	3pSg ⇒ Ved	(adds, added)
verb ⇒ V + able	(accept, acceptable)	verb ⇒ V + er	(begin, beginner)
verb ⇒ V + tion	(compile, compilation)	verb ⇒ V + ment	(agree, agreement)
adj ⇒ un + adj	(able, unable)	adj ⇒ adj + ly	(according, accordingly)
small ⇒ big	(brief, long)	thing ⇒ color	(ant, black)
thing ⇒ part	(bus, seats)	country ⇒ capital	(Austria, Vienna)
pronoun ⇒ possessive	(he, his)	male ⇒ female	(actor, actress)
lower ⇒ upper	(always, Always)	noun ⇒ plural	(album, albums)
adj ⇒ comparative	(bad, worse)	adj ⇒ superlative	(bad, worst)
frequent ⇒ infrequent	(bad, terrible)	English ⇒ French	(April, avril)
French ⇒ German	(ami, Freund)	French ⇒ Spanish	(année, año)
German ⇒ Spanish	(Arbeit, trabajo)		

Additional barplots The entire set of similarity barplots for the concepts of French–Spanish, French–German, English–French and German–Spanish are in Figures A.4, A.5,

Table A.6: Example context pairs from Winograd Schema

Contexts
The delivery truck zoomed by the school bus because it was going so fast. The delivery truck zoomed by the school bus because it was going so slow.
The man couldn't lift his son because he was so weak. The man couldn't lift his son because he was so heavy.
Joe's uncle can still beat him at tennis, even though he is 30 years younger. Joe's uncle can still beat him at tennis, even though he is 30 years older.
Paul tried to call George on the phone, but he wasn't successful. Paul tried to call George on the phone, but he wasn't available.
The large ball crashed right through the table because it was made of steel. The large ball crashed right through the table because it was made of styrofoam.

A.6 and A.7 respectively. As we see, they satisfy the same behavior as described earlier in Section 2.5.2, exhibiting relatively high similarity with the matching unembedding vector, close to high similarity with related language concepts and low similarity with unrelated concepts.

Table A.7: Top similarities between Winograd contexts and token concepts

CONTEXTS	MOST SIMILAR CONCEPT	SIMILARITY
Anne gave birth to a daughter last month. She is a very charming woman. Anne gave birth to a daughter last month. She is a very charming baby .	male⇒female	0.311
The table won't fit through the doorway because it is too wide. The table won't fit through the doorway because it is too narrow.	small⇒big	0.309
John couldn't see the stage with Billy in front of him because he is so short. John couldn't see the stage with Billy in front of him because he is so tall.	small⇒big	0.303

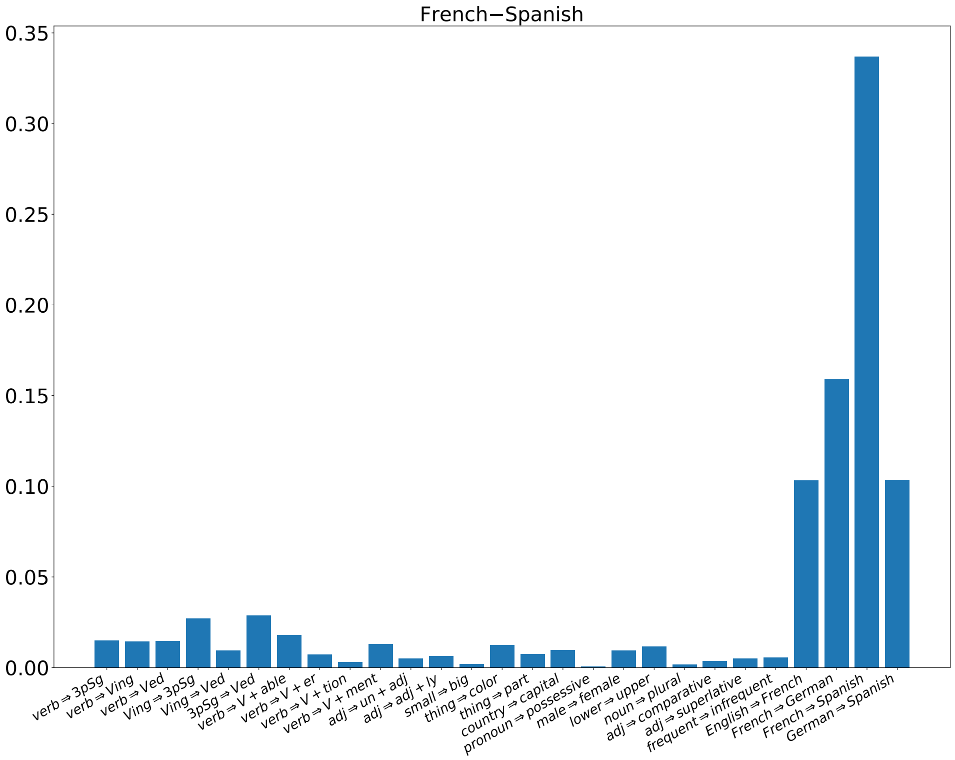


Figure A.4: The French–Spanish concept is highly correlated with similar token concepts relative to others. This figure shows all cosine similarities between the French–Spanish concept and token concepts.

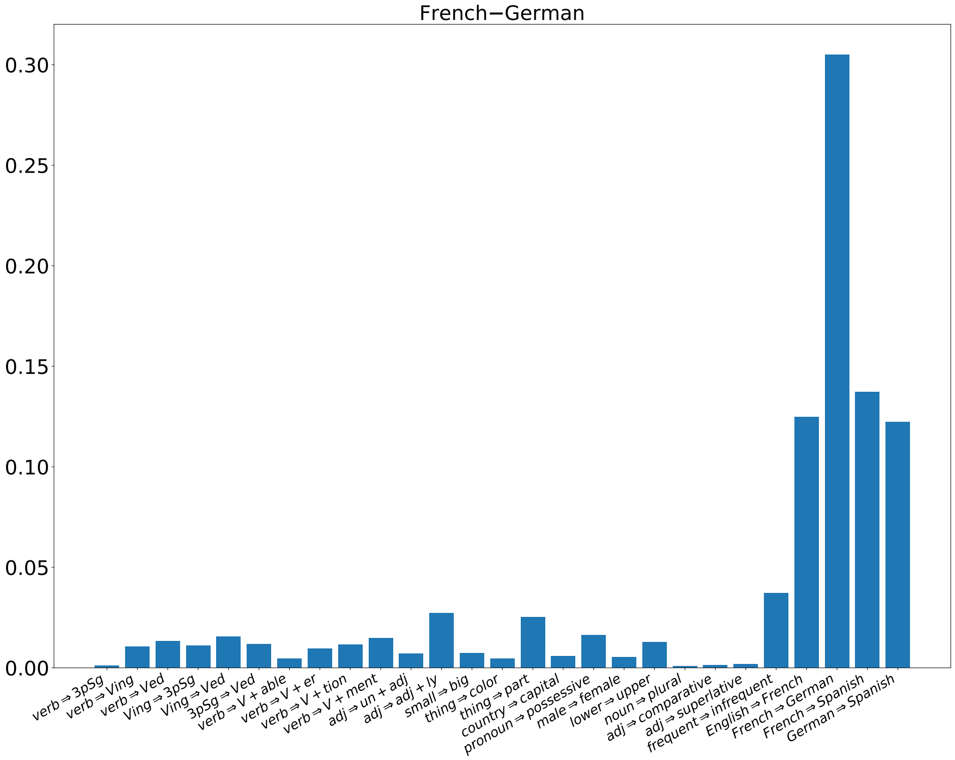


Figure A.5: The French–German concept is highly correlated with similar token concepts relative to others. This figure shows all cosine similarities between the French–German concept and token concepts.

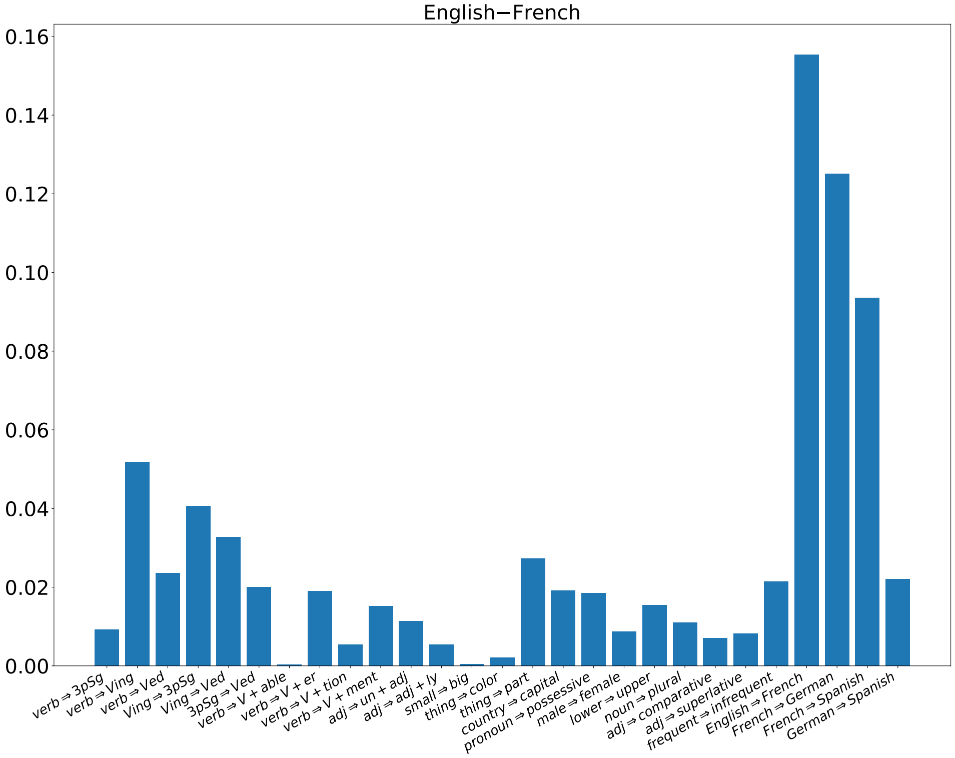


Figure A.6: The English–French concept is highly correlated with similar token concepts relative to others. This figure shows all cosine similarities between the English–French concept and token concepts.

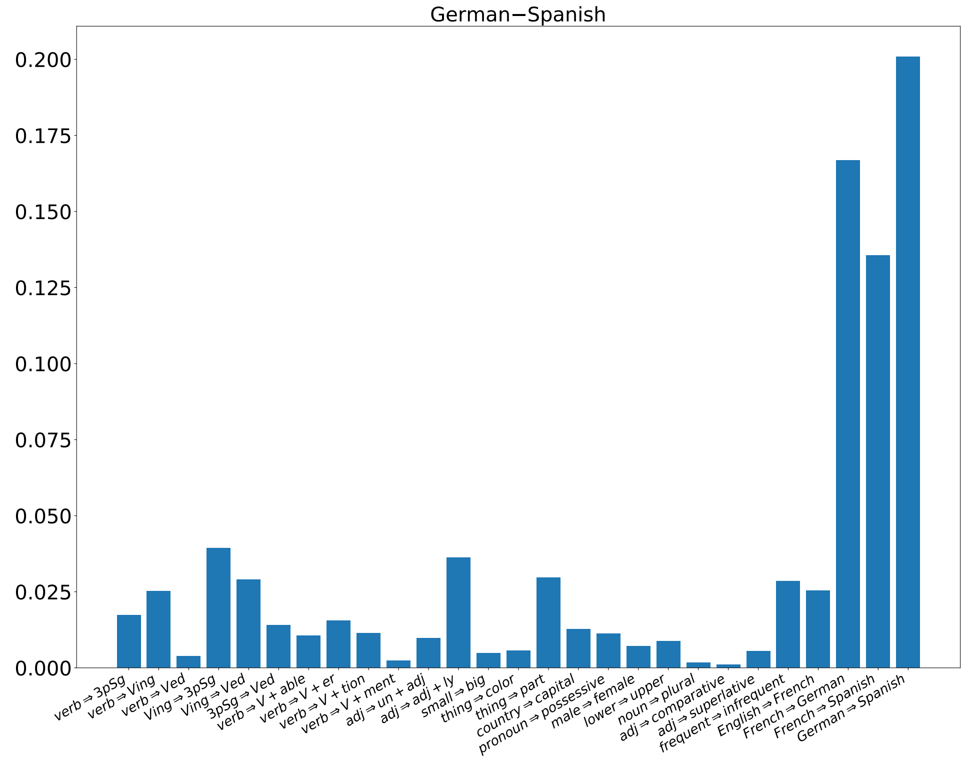


Figure A.7: The German–Spanish concept is highly correlated with similar token concepts relative to others. This figure shows all cosine similarities between the German–Spanish concept and token concepts.

APPENDIX B

APPENDIX FOR CHAPTER 3

B.1 Additional proofs

Lemma 35. *Let v be an element in* E*, and $\mathcal{M}$ a Markov boundary of v, then $\mathcal{M}$ is also a generalized Markov boundary.*

Proof. By definition, for any $u \in V \setminus (\{v\} \cup \mathcal{M})$, ${\mathrm{S^c}}_{\mathcal{M}}(v, u) = 0$. Therefore, $\mathcal{M}$ is also a generalized Markov boundary. □

Theorem 13. *Let partial orthogonality $\perp\!\!\!\perp_{\mathrm{O}}$ be the independence model over a finite set of embedding vectors* E*. Suppose $\mathcal{M}_1, \mathcal{M}_2 \subseteq$ E are two distinct Markov boundaries of $v_i \in$ E, then,*

$$proj_{\mathcal{M}_1}[v_i] = proj_{\mathcal{M}_2}[v_i]$$

Proof. To slightly abuse notation, we also use $\mathcal{M}_1$ to be a matrix where each column is an element in $\mathcal{M}_1$. We define $\mathcal{M}_2$ similarly.

Because $\mathcal{M}_1$ and $\mathcal{M}_2$ are two distinct Markov boundaries, they must not be empty. Therefore, $v_i \not\perp\!\!\!\perp_{\mathrm{O}} \mathcal{M}_1$ and $v_i \not\perp\!\!\!\perp_{\mathrm{O}} \mathcal{M}_2$. By the definition of Markov boundary, we also have $v_i \perp\!\!\!\perp_{\mathrm{O}} \mathcal{M}_1 \,|\, \mathcal{M}_2$ and $v_i \perp\!\!\!\perp_{\mathrm{O}} \mathcal{M}_2 \,|\, \mathcal{M}_1$. Note that $\mathcal{M}_2$ and $\mathcal{M}_1$ must have full rank, otherwise, they are not minimal.

Thus,

$$\begin{aligned}
&\langle \mathrm{proj}^{\perp}_{\mathcal{M}_1}[v_i],\ \mathrm{proj}^{\perp}_{\mathcal{M}_1}[v_j] \rangle = 0,\ \forall v_j \in \mathcal{M}_2 \\
\iff &\langle \mathrm{proj}^{\perp}_{\mathcal{M}_1}[v_i],\ v_j \rangle = 0,\ \forall v_j \in M_2 \\
\iff &v_i^T\, \mathcal{M}_1 (\mathcal{M}_1^T\, \mathcal{M}_1)^{-1}\, \mathcal{M}_1^T\, v_j = v_i^T v_j\ \forall v_j \in \mathcal{M}_2 \\
\iff &v_i^T\, \mathcal{M}_1 (\mathcal{M}_1^T\, \mathcal{M}_1)^{-1}\, \mathcal{M}_1^T\, \mathcal{M}_2 = v_i^T\, \mathcal{M}_2
\end{aligned}$$

With (compact) singular value decomposition, we have $\mathcal{M}_1 = U_1 \Sigma_1 V_1^T$ and $\mathcal{M}_2 = U_2 \Sigma_2 V_2^T$. Then,

$$\begin{aligned} v_i^T \mathcal{M}_1 (\mathcal{M}_1^T \mathcal{M}_1)^{-1} \mathcal{M}_1^T \mathcal{M}_2 = v_i^T U_1 U_1^T \mathcal{M}_2 = v_i^T \mathcal{M}_2 \\ \iff v_i^T U_1 U_1^T U_2 = v_i^T U_2 \end{aligned}$$

Similarly,

$$v_i^T U_2 U_2^T U_1 = v_i^T U_1$$

Therefore,

$$v_i^T U_1 U_1^T U_2 U_2^T = v_i^T U_2 U_2^T$$

In other words,

$$proj_{\mathcal{M}_2}[proj_{\mathcal{M}_1}[v_i]] = proj_{\mathcal{M}_2}[v_i]$$

On the other hand, $v_i \not\perp\!\!\!\perp_{\mathrm{O}} \mathcal{M}_1$.

$$proj_{\mathcal{M}_1}[v_i] = U_1 U_1^T v_i \neq 0$$

Similarly,

$$proj_{\mathcal{M}_2}[v_i] = U_2 U_2^T v_i \neq 0$$

Therefore, we must have,

$$proj_{\mathcal{M}_1}[v_i] \in \mathrm{span}(\mathcal{M}_2)$$

which means,

$$proj_{\mathcal{M}_2}[proj_{\mathcal{M}_1}[v_i]] = proj_{\mathcal{M}_1}[v_i] = proj_{\mathcal{M}_2}[v_i]$$

□

Corollary 14. *Let parital orthogonality $\perp\!\!\!\perp_{\mathrm{O}}$ be the independence model over a finite set of embedding vectors* E. *Suppose $\mathcal{M}_1 \subseteq$* E *is a Markov boundary of $v_i \in$* E *and $v_i \in$*

$\mathrm{span}(\mathrm{E} \setminus \{v_i\})$, *then,*

$$\mathit{proj}_{\mathcal{M}_1}[v_i] = v_i.$$

Proof. Because $v_i \in \mathrm{span}(\mathrm{E} \setminus \{v_i\})$, then $v_i = \sum_{k=1}^{m} \alpha_k v_k$ with $\mathrm{E}' = \{v_1, ..., v_m\} \subseteq \mathrm{E}$.

Since $\mathcal{M}_1$ is a Markov boundary of v_i,

$$v_i^T \mathcal{M}_1 (\mathcal{M}_1^T \mathcal{M}_1)^{-1} \mathcal{M}_1^T v_k = v_i^T v_k \; \forall v_k \in \mathrm{E}'$$

$$v_i^T \mathcal{M}_1 (\mathcal{M}_1^T \mathcal{M}_1)^{-1} \mathcal{M}_1^T \sum_{k=1}^{m} \alpha_k v_k = v_i^T \sum_{k=1}^{m} \alpha_k v_k$$
$$\langle \mathit{proj}_{\mathcal{M}_1}[v_i], v_i \rangle = \langle v_i, v_i \rangle$$

Thus, $\mathit{proj}_{\mathcal{M}_1}[v_i] = v_i$.

□

B.1.1 Construction of IPE map

Theorem 19. *Let V be a finite set of random variables with distribution P. $\mathcal{G}_P$ is a minimal I-map of P. Let A be equal to $\mathrm{adj}_\varepsilon(\mathcal{G}_P)^{-1}$ with eigen decomposition $A = U \Sigma U^T$. If ε is a perfect perturbation factor, then the function f with*

$$f(v_i) = U_i \Sigma^{1/2}$$

is an IPE map of P where v_i is a random variable in V and U_i is the i-th row of U. Furthermore, if P is faithful to $\mathcal{G}_P$, then f is a faithful IPE map for $\mathcal{P}$.

Proof. Let $|V| = n$ and, to slightly abuse notation, we use index $i \in [n]$ to mean the vertex v_i and the i-th embedding. And we use A_{V_1,V_2} where $V_1, V_2 \subseteq V$ to mean the submatrix of

$A_{\{i:v_i\in V_1\},\{i:v_i\in V_2\}}$.

Because G_p is a miminal I-map of P, we have $\mathcal{I}_G(V) \subseteq \mathcal{I}_P(V)$.

We just need to show $\mathcal{I}_G(V) = \mathcal{I}_O(f(V))$. And if P is faithful to G_p, then $\mathcal{I}_O(f(V)) = \mathcal{I}_G(V) = \mathcal{I}_P(V)$.

If $v_i \perp\!\!\!\perp_{\mathrm{G}} v_j | V'$ where $V' \subseteq V$, let $V^c = V \setminus V'$, then

$$A = \begin{pmatrix} A_{V^c}, A_{V^c,V'} \\ A_{V',V^c}, A_{V'} \end{pmatrix}$$

On the other hand, if $f(v_i) \perp\!\!\!\perp_{\mathrm{O}} f(v_j)|f(V')$, then

$$f(v_i)^T f(v_j) - f(v_i)^T f(V')(f(V')^T f(V'))^{-1} f(V')^T f(v_j) = 0 \tag{B.1.1}$$

Note that by our construction, $(f(V')^T f(V'))^{-1} = A_{V'}$ is invertible. We can write (B.1.1) as follows:

$$f(v_i)^T f(v_j) - f(v_i)^T f(V')(f(V')^T f(V'))^{-1} f(V')^T f(v_j) = A_{i,j} - A_{i,V'} A_{V'}^{-1} A_{V',j}$$

By Schur's complement, we have that

$$(\mathrm{adj}_{\varepsilon}(G_P)_{V^c})^{-1} = (A^{-1})_{V^c}^{-1} = A_{V^c} - A_{V^c,V'} A_{V'}^{-1} A_{V',V^c}$$

Becasue ε is a perfect perturbation factor, by definition, $f(v_i) \perp\!\!\!\perp_{\mathrm{O}} f(v_j)|f(V')$ if and only if $v_i \perp\!\!\!\perp_{\mathrm{G}} v_j | V'$. By the compositional property, we have that $\mathcal{I}_G(V) = \mathcal{I}_O(f(V))$. □

Lemma 21. *For any simple graph G, ε is perfect for all but finitely many $\varepsilon \in \mathbb{R}$.*

Proof. This is a direct consequence of Theorem 1 in Lněnička and Matúš [2007]. □

B.1.2 Dimension reduction of IPE

Theorem 22. *Let U be a set of vectors in $\mathbb{R}^n$ where $n = |U|$ and every vector is a unit vector. Let Σ be a matrix in $\mathbb{R}^{n\times n}$ where $\Sigma_{ij} = \langle u_i, u_j\rangle$. Assume $\lambda_1 = \lambda_{\min}(\Sigma) > 0$. Then there exists a mapping $g : \mathbb{R}^n \to \mathbb{R}^k$ where $k = \lceil 20\log(2n)/(\varepsilon')^2\rceil$ with $\varepsilon' = \min\{1/2, \varepsilon/C, \lambda_1/2r^2\}$ and $\varepsilon \in (0, 1)$ such that for any $u_i \in U$ with its unique Markov boundary $M_i \subseteq U$ and any $u_j \in U \setminus (\{u_i\} \cup M_i)$, we have*

$$\left|\left\langle \mathrm{proj}^{\perp}_{g(M_i)}[g(u_i)], \mathrm{proj}^{\perp}_{g(M_i)}[g(u_j)]\right\rangle\right| \leq \varepsilon$$

where $r_i = |M_i|$, $r = \max_i |M_i|$ and $C = (r+1)^3(\frac{2\lambda_{\max}(\Sigma)+2(r+1)^2}{\lambda_{\min}(\Sigma)})^r$.

Proof. Let g be linear map of Lemma 36 with error parameter $\varepsilon' \in (0, \frac{1}{2})$. For convenience, let $\tilde{u}_i = g(u_i)$, $\tilde{u_j} = g(u_j)$ and $\tilde{M}_i = g(M_i)$. Let $r_i = |M_i|$. To slightly abuse notation, we use M_i and $\tilde{M}_i$ to also mean matrices where each column is an element in the set. Furthermore, we also define $\tilde{\Sigma}$ to be $\tilde{\Sigma}_{ij} = \langle \tilde{u}_i, \tilde{u}_j\rangle$. We use $\Sigma_{A,B}$ where $A, B \subseteq U$ to be a submatrix where the row indices are from A and the column indices are from B, and when $A = B$, we just use Σ_A for simplicity. In particular, let $\Sigma_{(i,M_i),(j,M_i)} = \begin{pmatrix} \Sigma_{i,j}, & \Sigma_{i,M_i} \\ \Sigma_{M_i,j}, & \Sigma_{M_i} \end{pmatrix}$. And we can define a similar thing for $\tilde{\Sigma}$.

We first want to find ε' such that $\tilde{\Sigma}_{M_i}$ is non-singular for all $i \in |U|$. Note that for any $u_i \in U$, we know that by Weyl's inequality for eigenvalues [Horn and Johnson, 2012],

$$|\lambda_{\min}(\tilde{\Sigma}_{M_i}) - \lambda_{\min}(\Sigma_{M_i})| \leq ||\tilde{\Sigma}_{M_i} - \Sigma_{M_i}|| \leq ||\tilde{\Sigma}_{M_i} - \Sigma_{M_i}||_F \leq r^2\varepsilon'$$

Thus,

$$\lambda_{\min}(\tilde{\Sigma}_{M_i}) \geq \lambda_{\min}(\Sigma_{M_i}) - r^2\varepsilon \geq \lambda_{\min}(\Sigma) - r^2\varepsilon = \lambda_1 - r^2\varepsilon'$$

Therefore, if we want $\lambda_{\min}(\tilde{\Sigma}_{M_i}) > \frac{\lambda_1}{2}$ we need $\varepsilon' < \frac{\lambda_1}{2r^2}$.

On the other hand, because M_i is an Markov boundary for u_i, we have

$$u_i^T u_j - u_i^T M_i (M_i^T M_i)^{-1} M_i^T u_j = 0 \tag{B.1.2}$$

Note that M_i must be full rank. Otherwise, we can find a subset of M_i to be the Markov boundary. And there is a different way to write this. Remember that $\Sigma_{(i,M_i),(j,M_i)} = \begin{pmatrix} \Sigma_{i,j}, & \Sigma_{i,M_i} \\ \Sigma_{M_i,j}, & \Sigma_{M_i} \end{pmatrix}$. Using Schur's complement, we have that

$$\begin{aligned} \det(\Sigma_{(i,M_i),(j,M_i)}) &= \det(\Sigma_{M_i}) \det(\Sigma_{i,j} - \Sigma_{i,M_i}^T (\Sigma_{M_i})^{-1} \Sigma_{j,M_i}) \\ &= \det(\Sigma_{M_i})(u_i^T u_j - u_i^T M_i (M_i^T M_i)^{-1} M_i^T u_j) \end{aligned}$$

We want to estimate the following:

$$\begin{aligned} |\langle \text{proj}_{g(M_i)}^{\perp}[g(u_i)], \text{proj}_{g(M_i)}^{\perp}[g(u_j)] \rangle| &= |\tilde{u_i}^T \tilde{u_j} - \tilde{u_i}^T \tilde{M_i} (\tilde{M_i}^T \tilde{M_i})^{-1} \tilde{M_i}^T \tilde{u_j}| \\ &= |\frac{\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)})}{\det(\tilde{\Sigma}_{M_i})}| \end{aligned} \tag{B.1.3}$$

We already know that $\det(\tilde{\Sigma}_{M_i}) > (\frac{\lambda_1}{2})^{r_i}$. On the other hand, by Theorem 2.12 in Ipsen and Rehman [2008], we have that

$$\begin{aligned} &|\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)})| = |\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)}) - \det(\Sigma_{(i,M_i),(j,M_i)})| \\ &\leq (r_i + 1)||\tilde{\Sigma}_{(i,M_i),(j,M_i)} - \Sigma_{(i,M_i),(j,M_i)}|| \max\{||\Sigma_{(i,M_i),(j,M_i)}||, ||\tilde{\Sigma}_{(i,M_i),(j,M_i)}||\}^{r_i} \end{aligned}$$

By Weyl's inequality for singular values, we have that

$$\begin{aligned} ||\tilde{\Sigma}_{(i,M_i),(j,M_i)}|| = \sigma_{\max}(\tilde{\Sigma}_{(i,M_i),(j,M_i)}) &\leq \sigma_{\max}(\Sigma_{(i,M_i),(j,M_i)}) + (r_i + 1)^2 \varepsilon' \\ &\leq \lambda_{\max}(\Sigma) + (r_i + 1)^2 \varepsilon' \leq \lambda_{\max}(\Sigma) + (r_i + 1)^2 \end{aligned}$$

Thus,

$$|\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)})| \leq (r_i+1)(r_i+1)^2\varepsilon'(\lambda_{\max}(\Sigma)+(r_i+1)^2)^{r_i}$$

And,

$$|\frac{\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)})}{\det(\tilde{\Sigma}_{M_i})}| \leq \varepsilon' \frac{(r_i+1)^3(\lambda_{\max}(\Sigma)+(r_i+1)^2)^{r_i}}{(\frac{\lambda_1}{2})^{r_i}}$$

Let $C = (r+1)^3(\frac{2\lambda_{\max}(\Sigma)+2(r+1)^2}{\lambda_{\min}(\Sigma)})^r$. Then,

$$|\frac{\det(\tilde{\Sigma}_{(i,M_i),(j,M_i)})}{\det(\tilde{\Sigma}_{M_i})}| \leq \varepsilon' C$$

Let $\varepsilon' = \min\{\frac{1}{2}, \frac{\varepsilon}{C}, \frac{\lambda_1}{2r^2}\}$ and $k = \left\lceil \frac{20\log(2n)}{(\varepsilon')^2} \right\rceil$, we have that

$$|\langle \text{proj}^{\perp}_{g(M_i)}[g(u_i)], \text{proj}^{\perp}_{g(M_i)}[g(u_j)] \rangle| \leq \varepsilon$$

□

Lemma 36. *Let $\varepsilon \in (0, \frac{1}{2})$. Let $V \subseteq \mathbb{R}^d$ be a set of n points and $k = \left\lceil \frac{20\log(2n)}{\varepsilon^2} \right\rceil$, there exists a linear mapping $g : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in V$:*

$$|\langle g(u), g(v) \rangle - \langle u, v \rangle| \leq \varepsilon$$

Proof. The proof is an easy extension of the JL lemma [Vempala, 2005] by adding all the $-v_i$ for all $v_i \in V$ into the set V. □

B.2 Additional experiments, figures and tables

Table B.1: 9 categories of words used to test the semantic meaning of partial orthogonality

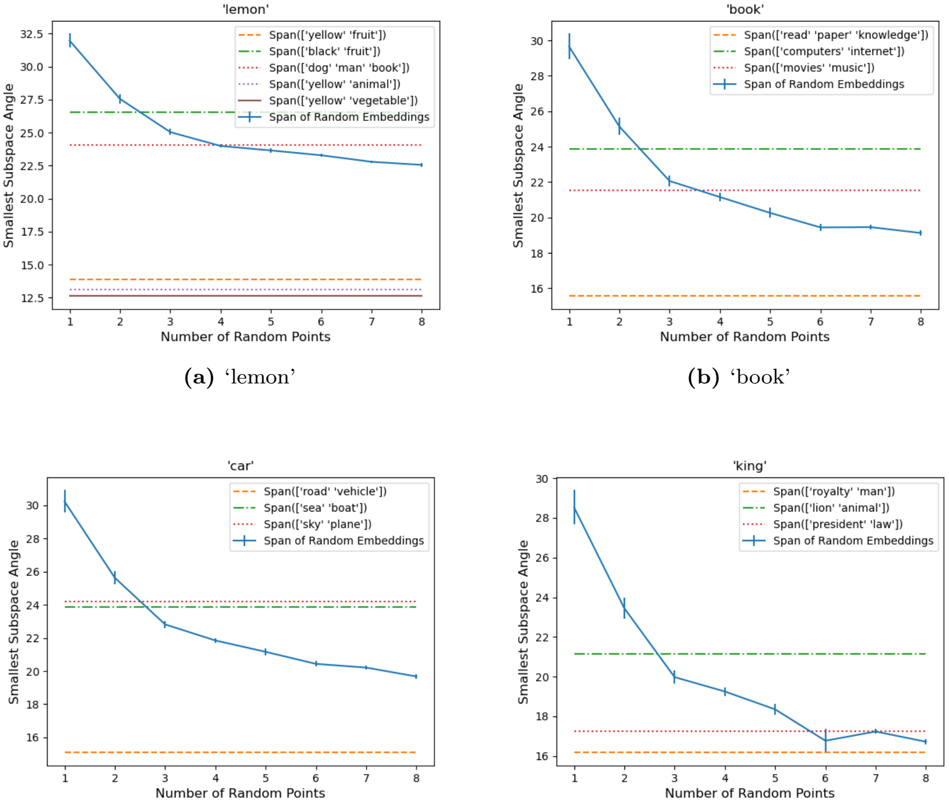
Category	Words in Category
‘vehicle’	‘car’, ‘bicycle’, ‘skateboard’, ‘motorcycle’, ‘helicopter’, ‘truck’, ‘boat’, ‘airplane’, ‘submarine’, ‘scooter’
‘animal’	‘lion’, ‘dolphin’, ‘eagle’, ‘dog’, ‘elephant’, ‘cat’, ‘rat’, ‘giraffe’, ‘bird’, ‘tiger’
‘tool’	‘hammer’, ‘screwdriver’, ‘wrench’, ‘pliers’, ‘hacksaw’, ‘drill’, ‘chisel’, ‘plunger’, ‘trowel’, ‘cutter’
‘clothing’	‘shirt’, ‘pants’, ‘dress’, ‘sweater’, ‘jacket’, ‘hat’, ‘socks’, ‘gloves’, ‘scarf’, ‘vest’
‘beverage’	‘coffee’, ‘tea’, ‘soda’, ‘lemonade’, ‘milk’, ‘wine’, ‘beer’, ‘sake’, ‘smoothie’, ‘nectar’
‘science’	‘biology’, ‘ecology’, ‘genetics’, ‘chemistry’, ‘physics’, ‘geology’, ‘mathematics’, ‘linguistics’, ‘psychology’, ‘cryptography’
‘furniture’	‘couch’, ‘bed’, ‘cabinet’, ‘dresser’, ‘hallstand’, ‘lamp’, ‘bench’, ‘chair’, ‘table’, ‘closet’
‘plant’	‘daisy’, ‘pine’, ‘iris’, ‘lily’, ‘oak’, ‘tulip’, ‘fern’, ‘rose’, ‘bamboo’, ‘cactus’
‘food’	‘chocolate’, ‘meat’, ‘steak’, ‘pasta’, ‘fish’, ‘brisket’, ‘sausage’, ‘loaf’, ‘roe’, ‘lobster’

Table B.2: Experiments show that top correlated words with target words after projecting onto the orthogonal complements of randomly selected linear subspaces are more semantically meaningful

Target	Top Correlated Words Before Projection	Top Correlated Words After Projection
‘eggplant’	‘potato’, ‘banana’, ‘grape’, ‘vegetable’, ‘bananas’, ‘tomato’, ‘espagnol’, ‘eternal’, ‘potatoes’, ‘e.g.’	‘grape’, ‘purple-black’, ‘purple’, ‘turnips’, ‘plum’, ‘lilac’, ‘vegetable’, ‘vegetables’, ‘banana’, ‘ultra-violet’
‘king’	‘mister’, ‘bossman’, ‘thet’, ‘thatt’, ‘beast’, ‘killed’, ‘yesiree’, ‘bossed’, ‘outdo’, ‘queen’s’	‘royalty’, ‘sport-king’, ‘bossman’, ‘kingan’, ‘mister’, ‘prince’s’, ‘princess’, ‘princes’, ‘handsomest’, ‘ruling’
‘advise’	‘spoken’, ‘askin’, ‘concur’, ‘applies’, ‘said’, ‘according’, ‘astute’, ‘pertinent’, ‘evident’, ‘preached’	‘guidelines’, ‘guidance’, ‘tips’, ‘motto’, ‘motivating’, ‘encourages’, ‘advising’, ‘advisory’, ‘self-help’, ‘reminder’
‘work-out’	‘healthy’, ‘weights’, ‘worked’, ‘time-on-the-job’, ‘on-the-job’, ‘work-success’, ‘busy-work’, ‘out’n’, ‘healthiest’, ‘hardworking’	‘gym’, ‘weights’, ‘footing’, ‘running’, ‘jogs’, ‘dumbbells’, ‘conditioning’, ‘bodybuilding’, ‘runing’, ‘pumped-up’
‘poem’	‘!’, ‘ya’, ‘eh’, ‘yes’, ‘;’, ‘mem’, ‘oh’, ‘)’, ‘poignant’, ‘hee’	‘poems’, ‘poetizing’, ‘poetry’s’, ‘rhyming’, ‘sonnet’, ‘lyrics’, ‘recited’, ‘poetically’, ‘sonnets’, ‘rhyme’

Table B.3: Selected examples provided by ChatGPT when asked "give me a list of 50 common nouns, each with a short description, and the first one is eggplant"

Target Word	Description Sentence
'eggplant'	'A purple or dark-colored vegetable with a smooth skin, often used in cooking and known for its mild flavor.'
'dog'	'A domesticated mammal often kept as a pet or used for various purposes.'
'book'	'A physical or digital publication containing written or printed content.'
'car'	'A motorized vehicle used for transportation on roads.'
'tree'	'A woody perennial plant with a main trunk and branches, usually producing leaves.'
'house'	'A building where people live, providing shelter and accommodation.'
'computer'	'An electronic device used for processing and storing data, and performing various tasks.'
'cat'	'A small domesticated carnivorous mammal commonly kept as a pet.'
'chair'	'A piece of furniture designed for sitting on, often with a backrest and four legs.'
'phone'	'A communication device that allows voice calls and text messaging.'



(a) ‘lemon’

(b) ‘book’

(c) ‘car’

(d) ‘king’

Figure B.1: Linear subspaces spanned by estimated generalized Markov boundaries have smallest subspace angles with linear subspaces spanned by embeddings that best match semantic meanings

APPENDIX C

APPENDIX FOR CHAPTER 4

C.1 Additional theoretical results and proofs

C.1.1 Proofs for Section 4.4.1

Theorem 24 can be stated more formally as follows:

Theorem 37. *Suppose the data generating process follows Section 4.3.1 where $m \geq 3$, $\omega = 1$, and $\mathcal{N}(t) = V \setminus \{t\}$. Assume there exists a single layer transformer given by (4.3.1) such that a) $W_K = 0$ and $W_Q = 0$, b) Each row of W_E is orthogonal to each other and normalized, and c) W_V is given by*

$$W_V = \sum_{i \in [V]} W_E(i)(\sum_{j \in \mathcal{N}_1(i)} W_E(j)^T).$$

Then if $L > \max\{\frac{100m^2 \log(3/\varepsilon)}{(\exp(-\frac{1}{\beta})-\exp(-\frac{2}{\beta}))^2}, \frac{80m^2 |\mathcal{N}(y)|}{(\exp(-\frac{1}{\beta})-\exp(-\frac{2}{\beta}))^2}\}$ for any y, then

$$R_{\mathcal{D}^L}(f^L) \leq \varepsilon,$$

where $0 < \varepsilon < 1$.

Proof. First of all, the error is defined to be:

$$\begin{aligned} R_{\mathcal{D}^L}(f^L) &= \mathbb{P}_{(x,y)\sim\mathcal{D}^L}[\operatorname{argmax} f^L(x) \neq y] \\ &= \mathbb{P}_y \mathbb{P}_{x|y}[\operatorname{argmax} f^L(x) \neq y] \end{aligned}$$

Let's focus on the conditional probability $\mathbb{P}_{x|y}[\operatorname{argmax} f^L(x) \neq y]$.

By construction, the single layer transformer model has uniform attention. Therefore,

$$h(x) = \sum_{i \in \mathcal{N}(y)} \alpha_i W_E(i)$$

where $\alpha_i = \frac{1}{L}\sum_{k=1}^{L} \mathbf{1}\{t_k = i\}$ which is the number of occurrence of token i in the sequence.

By the latent concept association model, we know that

$$p(i|y) = \frac{\exp(-D_H(i,y)/\beta)}{Z}$$

where $Z = \sum_{i \in \mathcal{N}(y)} \exp(-D_H(i,y)/\beta)$.

Thus, the logit for token y is

$$f_y^L(x) = \sum_{i \in \mathcal{N}_1(y)} \alpha_i$$

And the logit for any other token $\tilde{y}$ is

$$f_{\tilde{y}}^L(x) = \sum_{i \in \mathcal{N}_1(\tilde{y})} \alpha_i$$

For the prediction to be correct, we need

$$\max_{\tilde{y}} f_y^L(x) - f_{\tilde{y}}^L(x) > 0$$

By Lemma 3 of Devroye [1983], we know that for all $\Delta \in (0,1)$, if $\frac{|\mathcal{N}(y)|}{L} \leq \frac{\Delta^2}{20}$, we have

$$\mathbb{P}\big(\max_{i \in \mathcal{N}(y)} |\alpha_i - p(i|y)| > \Delta\big) \leq \mathbb{P}\big(\sum_{i \in \mathcal{N}(y)} |\alpha_i - p(i|y)| > \Delta\big) \leq 3\exp(-L\Delta^2/25)$$

Therefore, if $L \geq \max\{\frac{25\log(3/\varepsilon)}{\Delta^2}, \frac{20|\mathcal{N}(y)|}{\Delta^2}\}$, then with probability at least $1-\varepsilon$, we have,

$$\max_{i \in \mathcal{N}(y)} |\alpha_i - p(i|y)| \leq \Delta$$

$$
\begin{aligned}
f_y^L(x) - f_{\tilde{y}}^L(x) &= \sum_{i \in \mathcal{N}_1(y)} \alpha_i - \sum_{j \in \mathcal{N}_1(\tilde{y})} \alpha_j \\
&= \sum_{i \in \mathcal{N}_1(y)} \alpha_i - \sum_{i \in \mathcal{N}_1(y)} p(i|y) + \sum_{i \in \mathcal{N}_1(y)} p(i|y) \\
&\quad - \sum_{j \in \mathcal{N}_1(\tilde{y})} p(j|y) + \sum_{j \in \mathcal{N}_1(\tilde{y})} p(j|y) - \sum_{j \in \mathcal{N}_1(\tilde{y})} \alpha_j \\
&\geq \sum_{i \in \mathcal{N}_1(y)} p(i|y) - \sum_{j \in \mathcal{N}_1(\tilde{y})} p(j|y) - 2m\Delta \\
&\geq \exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}) - 2m\Delta
\end{aligned}
$$

Note that because of Lemma 40, there's no neighboring set that is the superset of another.

Therefore as long as $\Delta < \frac{\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta})}{2m}$,

$$
f_y^L(x) - f_{\tilde{y}}^L(x) > 0
$$

for any $\tilde{y}$.

Finally, if $L > \max\{\frac{100m^2 \log(3/\varepsilon)}{(\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}))^2}, \frac{80m^2 |\mathcal{N}(y)|}{(\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}))^2}\}$ for any y, then

$$
\mathbb{P}_{x|y}[\operatorname{argmax} f^L(x) \neq y] \leq \varepsilon
$$

And

$$
\begin{aligned}
R_{\mathcal{D}^L}(f^L) &= \mathbb{P}_{(x,y)\sim\mathcal{D}^L}[\operatorname{argmax} f^L(x) \neq y] \\
&= \mathbb{P}_y \mathbb{P}_{x|y}[\operatorname{argmax} f^L(x) \neq y] \leq \varepsilon
\end{aligned}
$$

□

C.1.2 Proofs for Section 4.4.2

Lemma 25. *Suppose the data generating process follows Section 4.3.1 where* $m \geq 3$*,* $\omega = 1$ *and* $\mathcal{N}(t) = \{t' : D_H(t, t')) = 1\}$*. For any single layer transformer given by (4.3.1) where each row of* W_E *is orthogonal to each other and normalized, if* W_V *is constructed as in (4.4.1), then the error rate is* 0*. If* W_V *is the identity matrix, then the error rate is strictly larger than* 0*.*

Proof. Following the proof for Theorem 37, let's focus on the conditional probability:

$$\mathbb{P}_{x|y}[\text{argmax}\, f^L(x) \neq y]$$

By construction, we have

$$h(x) = \sum_{i \in \mathcal{N}_1(y)} \alpha_i W_E(i)$$

where $\alpha_i = \frac{1}{L}\sum_{k=1}^{L} \mathbf{1}\{t_k = i\}$ which is the number of occurrence of token i in the sequence.

Let's consider the first case where W_V is constructed as in (4.4.1). Then we know that for some other token $\tilde{y} \neq y$,

$$f^L_y(x) - f^L_{\tilde{y}}(x) = \sum_{i \in \mathcal{N}_1(y)} \alpha_i - \sum_{i \in \mathcal{N}_1(\tilde{y})} \alpha_i = 1 - \sum_{i \in \mathcal{N}_1(\tilde{y})} \alpha_i$$

By Lemma 40, we have that for any token $\tilde{y} \neq y$,

$$f^L_y(x) - f^L_{\tilde{y}}(x) > 0$$

Therefore, the error rate is always 0.

Now let's consider the second case where W_V is the identity matrix. Let j be a token in the set $\mathcal{N}_1(y)$. Then there is a non-zero probability that context x contains only j. In that case,

$$h(x) = W_E(j)$$

However, we know that by the assumption on the embedding matrix,

$$f_y^L(x) - f_j^L(x) = (W_E(y) - W_E(j))^T h(x) = -\|W_E(j)\|^2 < 0$$

This implies that there's non zero probability that y is misclassified. Therefore, when W_V is the identity matrix, the error rate is strictly larger than 0. □

Theorem 26. *Suppose the data generating process follows Section 4.3.1 where $m \geq 3$, $\omega = 1$ and $\mathcal{N}(t) = V \setminus \{t\}$. For any single layer transformer given by (4.3.1) with W_V being the identity matrix, if the cross entropy loss is minimized so that for any sampled pair (x, y),*

$$p(y|x) = \hat{p}(y|x) = \mathit{softmax}(f_y^L(x))$$

there exists $a > 0$ and b such that for two tokens $t \neq t'$,

$$\langle W_E(t), W_E(t') \rangle = -aD_H(t, t') + b$$

Proof. Because for any pair of (x, y), the estimated conditional probability matches the true conditional probability. In particular, let's consider two target tokens y_1, y_2 and context $x = (t_i, ..., t_i)$ for some token t_i such that $p(x|y_1) > 0$ and $p(x|y_2) > 0$, then

$$\frac{p(y_1|x)}{p(y_2|x)} = \frac{p(x|y_1)p(y_1)}{p(x|y_2)p(y_2)} = \frac{p(x|y_1)}{p(x|y_2)} = \frac{\hat{p}(x|y_1)}{\hat{p}(x|y_2)} = \exp((W_E(y_1) - W_E(y_2))^T h(x))$$

The second equality is because $p(y)$ is the uniform distribution. By our construction,

$$\frac{p(x|y_1)}{p(x|y_2)} = \frac{p(t_i|y_1)^L}{p(t_i|y_2)^L} = \exp((W_E(y_2) - W_E(y_1))^T h(x)) = \exp((W_E(y_1) - W_E(y_2))^T W_E(t_i))$$

By the data generating process, we have that

$$\frac{L}{\beta}(D_H(t_i, y_2) - D_H(t_i, y_1)) = (W_E(y_1) - W_E(y_2))^T W_E(t_i)$$

Let $t_i = y_3$ such that $y_3 \neq y_1$, $y_3 \neq y_2$, then

$$\frac{L}{\beta} D_H(y_3, y_1) - W_E(y_1)^T W_E(y_3) = \frac{L}{\beta} D_H(y_3, y_2) - W_E(y_2)^T W_E(y_3)$$

For simplicity, let's define

$$\Psi(y_1, y_2) = \frac{L}{\beta} D_H(y_1, y_2) - W_E(y_1)^T W_E(y_2)$$

Therefore,

$$\Psi(y_3, y_1) = \Psi(y_3, y_2)$$

Now consider five distinct labels: y_1, y_2, y_3, y_4, y_5. We have,

$$\Psi(y_3, y_1) = \Psi(y_3, y_2) = \Psi(y_4, y_2) = \Psi(y_4, y_5)$$

In other words, $\Psi(y_3, y_1) = \Psi(y_4, y_5)$ for arbitrarily chosen distinct labels y_1, y_3, y_4, y_5. Therefore, $\Psi(t, t')$ is a constant for $t \neq t'$.

For any two tokens $t \neq t'$,

$$\frac{L}{\beta} D_H(t, t') - W_E(t)^T W_E(t') = C$$

Thus,

$$W_E(t)^T W_E(t') = -\frac{L}{\beta} D_H(t, t') + C$$

□

C.1.3 Proofs for Section 4.4.3

Theorem 27 can be formalized as the following theorem.

Theorem 38. *Following the same setup as in Theorem 37, but embeddings follow (4.4.2) then if $b > 0$, $\Delta_1 > 0$, $0 < \Delta < \frac{\exp(-\frac{1}{\beta})-\exp(-\frac{2}{\beta})}{2m}$, $L \geq \max\{\frac{25\log(3/\varepsilon)}{\Delta^2}, \frac{20|\mathcal{N}(y)|}{\Delta^2}\}$ for any y, and*

$$0 < a < \frac{2\exp(\frac{1}{\beta})}{(|V|-2)m^2}$$

and

$$b_0 > \max\{\frac{a(m-2)m+\Delta_1}{\exp(-\frac{1}{\beta})-\exp(-\frac{2}{\beta})-2m\Delta}+b, \frac{(b-a)\Delta_1 - \frac{|V|-2}{2}abm^2\exp(-\frac{1}{\beta})+\frac{|V|-2}{2}a^2(m-2)m^2}{1-\frac{|V|-2}{2}am^2\exp(-\frac{1}{\beta})}\}$$

we have

$$R_{\mathcal{D}^L}(f^L) \leq \varepsilon$$

where $0 < \varepsilon < 1$.

Proof. Following the proof of Theorem 37, let's also focus on the conditional probability

$$\mathbb{P}_{x|y}[\operatorname{argmax} f^L(x) \neq y]$$

By construction, the single layer transformer model has uniform attention. Therefore,

$$h(x) = \sum_{i\in\mathcal{N}(y)} \alpha_i W_E(i)$$

where $\alpha_i = \frac{1}{L}\sum_{k=1}^{L}\mathbf{1}\{t_k = i\}$ which is the number of occurrence of token i in the sequence. For simplicity, let's define $\alpha_y = 0$ such that

$$h(x) = \sum_{i\in[V]} \alpha_i W_E(i)$$

Similarly, we also have that if $L \geq \max\{\frac{25\log(3/\varepsilon)}{\Delta^2}, \frac{20|\mathcal{N}(y)|}{\Delta^2}\}$, then with probability at least $1-\varepsilon$, we have,

$$\max_{i\in[V]} |\alpha_i - p(i|y)| \leq \Delta$$

Also define the following:

$$\begin{aligned}\phi_k(x) &= \sum_{j\in\mathcal{N}_1(k)} W_E(j)^T \big(\sum_{i\in[V]} \alpha_i W_E(i)\big)\\ v_k(y) &= W_E(y)^T W_E(k)\end{aligned}$$

Thus, the logit for token y is

$$f_y^L(x) = \sum_{k=0}^{|V|-1} v_k(y)\phi_k(x)$$

Let's investigate $\phi_k(x)$. By Lemma 39,

$$\begin{aligned}\phi_k(x) &= \sum_{i\in[V]} \alpha_i(\sum_{j\in\mathcal{N}_1(k)} W_E(j)^T W_E(i))\\ &= (b_0 - b)\sum_{j\in\mathcal{N}_1(k)} \alpha_j + \sum_{i\in[V]} \alpha_i(-a(m-2)D_H(k,i) + (b-a)m)\end{aligned}$$

Thus, for any $k_1, k_2 \in [V]$,

$$\begin{aligned}\phi_{k_1}(x) - \phi_{k_2}(x) = (b_0 - b)(&\sum_{j_1\in\mathcal{N}_1(k_1)} \alpha_{j_1} - \sum_{j_2\in\mathcal{N}_1(k_2)} \alpha_{j_2})\\ &+ \sum_{i\in[V]} \alpha_i a(m-2)(D_H(k_2,i) - D_H(k_1,i))\end{aligned}$$

Because $-m \leq D_H(k_2, i) - D_H(k_1, i) \leq m$, we have

$$(b_0 - b)(\sum_{j_1 \in \mathcal{N}_1(k_1)} \alpha_{j_1} - \sum_{j_2 \in \mathcal{N}_1(k_2)} \alpha_{j_2}) - a(m-2)m$$
$$\leq \phi_{k_1}(x) - \phi_{k_2}(x) \leq$$
$$(b_0 - b)(\sum_{j_1 \in \mathcal{N}_1(k_1)} \alpha_{j_1} - \sum_{j_2 \in \mathcal{N}_1(k_2)} \alpha_{j_2}) + a(m-2)m$$

For prediction to be correct, we need

$$\max_{\tilde{y}} f_y^L(x) - f_{\tilde{y}}^L(x) > 0$$

This also means that

$$\max_{\tilde{y}} \sum_{k=0}^{|V|-1} \big(v_k(y) - v_k(\tilde{y})\big)\phi_k(x) > 0$$

One can show that for any k, if $\iota^{-1}(\tilde{k}) = \iota^{-1}(y) \otimes \iota^{-1}(\tilde{y}) \otimes \iota^{-1}(k)$ where $\otimes$ means bitwise XOR, then

$$v_k(y) - v_k(\tilde{y}) = v_{\tilde{k}}(\tilde{y}) - v_{\tilde{k}}(y) \tag{C.1.1}$$

First of all, if $k = y$, then $\tilde{k} = \tilde{y}$, which means

$$v_k(y) - v_k(\tilde{y}) = v_{\tilde{k}}(\tilde{y}) - v_{\tilde{k}}(y) = b_0 + aD_H(y, \tilde{y}) - b$$

If $k \neq y, \tilde{y}$, then (C.1.1) implies that

$$D_H(k, y) - D_H(k, \tilde{y}) = D_H(\tilde{k}, \tilde{y}) - D_H(\tilde{k}, y)$$

We know that $D_H(k, y)$ is the number of 1s in $\iota^{-1}(k) \otimes \iota^{-1}(y)$ and,

$$\iota^{-1}(\tilde{k}) \otimes \iota^{-1}(y) = \iota^{-1}(y) \otimes \iota^{-1}(\tilde{y}) \otimes \iota^{-1}(k) \otimes \iota^{-1}(y) = \iota^{-1}(\tilde{y}) \otimes \iota^{-1}(k)$$

Similarly,

$$\iota^{-1}(\tilde{k}) \otimes \iota^{-1}(\tilde{y}) = \iota^{-1}(y) \otimes \iota^{-1}(k)$$

Therefore, (C.1.1) holds and we can rewrite $f_y^L(x) - f_{\tilde{y}}^L(x)$ as

$$\begin{aligned} f_y^L(x) - f_{\tilde{y}}^L(x) &= \sum_{k=0}^{|V|-1} \big(v_k(y) - v_k(\tilde{y})\big)\phi_k(x) \\ &= (b_0 - b + aD_H(y,\tilde{y}))(\phi_y(x) - \phi_{\tilde{y}}(x)) \\ &+ \sum_{k \neq y, \tilde{y}, D_H(k,y) \geq D_H(k,\tilde{y})} a(D_H(k,y) - D_H(k,\tilde{y}))(\phi_k(x) - \phi_{\tilde{k}}(x)) \end{aligned}$$

We already know that $b_0 > b > 0$ and $a > 0$, thus, $b_0 - b + aD_H(y,\tilde{y}) > 0$ for any pair $y, \tilde{y}$.

We also want $\phi_y(x) - \phi_{\tilde{y}}(x)$ to be positive. Note that

$$\phi_y(x) - \phi_{\tilde{y}}(x) \geq (b_0 - b)(\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}) - 2m\Delta) - a(m-2)m$$

We need $\Delta < \frac{\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta})}{2m}$ and for some positive $\Delta_1 > 0$, b_0 needs to be large enough such that

$$\phi_y(x) - \phi_{\tilde{y}}(x) > \Delta_1$$

which implies that

$$b_0 > \frac{a(m-2)m + \Delta_1}{\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}) - 2m\Delta} + b \tag{C.1.2}$$

On the other hand, for $k \neq y, \tilde{y}$, we have

$$\begin{aligned}
\phi_k(x) - \phi_{\tilde{k}}(x) &\geq (b_0 - b)(\sum_{j_1 \in \mathcal{N}_1(k)} \alpha_{j_1} - \sum_{j_2 \in \mathcal{N}_1(\tilde{k})} \alpha_{j_2}) - a(m-2)m \\
&\geq (b_0 - b)(-(m-1)\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}) - 2m\Delta) - a(m-2)m \\
&\geq (b_0 - b)(-(m-1)\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta}) + \exp(-\frac{2}{\beta}) - \exp(-\frac{1}{\beta})) - a(m-2)m \\
&\geq -(b_0 - b)m\exp(-\frac{1}{\beta}) - a(m-2)m
\end{aligned}$$

Then, we have

$$\begin{aligned}
&f_y^L(x) - f_{\tilde{y}}^L(x) \geq (b_0 - b + a)\Delta_1 - \frac{|V|-2}{2}\bigg((b_0 - b)am^2\exp(-\frac{1}{\beta}) + a^2(m-2)m^2\bigg) \\
&\geq \bigg(1 - \frac{|V|-2}{2}am^2\exp(-\frac{1}{\beta})\bigg)b_0 - (b-a)\Delta_1 + \frac{|V|-2}{2}abm^2\exp(-\frac{1}{\beta}) - \frac{|V|-2}{2}a^2(m-2)m^2
\end{aligned}$$

The lower bound is independent of $\tilde{y}$, therefore, we need it to be positive to ensure the prediction is correct. To achieve this, we want

$$1 - \frac{|V|-2}{2}am^2\exp(-\frac{1}{\beta}) > 0$$

which implies that

$$a < \frac{2\exp(\frac{1}{\beta})}{(|V|-2)m^2} \tag{C.1.3}$$

And finally we need

$$b_0 > \frac{(b-a)\Delta_1 - \frac{|V|-2}{2}abm^2\exp(-\frac{1}{\beta}) + \frac{|V|-2}{2}a^2(m-2)m^2}{1 - \frac{|V|-2}{2}am^2\exp(-\frac{1}{\beta})} \tag{C.1.4}$$

To summarize, if $b > 0$, $\Delta_1 > 0$, $0 < \Delta < \frac{\exp(-\frac{1}{\beta}) - \exp(-\frac{2}{\beta})}{2m}$, $L \geq \max\{\frac{25\log(3/\varepsilon)}{\Delta^2}, \frac{20|\mathcal{N}(y)|}{\Delta^2}\}$

for any y, and

$$0 < a < \frac{2\exp(\frac{1}{\beta})}{(|V|-2)m^2}$$

and

$$b_0 > \max\{\frac{a(m-2)m+\Delta_1}{\exp(-\frac{1}{\beta})-\exp(-\frac{2}{\beta})-2m\Delta}+b, \frac{(b-a)\Delta_1 - \frac{|V|-2}{2}abm^2\exp(-\frac{1}{\beta})+\frac{|V|-2}{2}a^2(m-2)m^2}{1-\frac{|V|-2}{2}am^2\exp(-\frac{1}{\beta})}\}$$

we have

$$R_{\mathcal{D}^L}(f^L) \leq \varepsilon$$

where $0 < \varepsilon < 1$.

□

Lemma 28. *If embeddings follow (4.4.2) and $b = b_0$ and $\mathcal{N}(t) = V \setminus \{t\}$, then rank($W_E$) $\leq$ $m+2$.*

Proof. By (4.4.2), we have that

$$\langle W_E(i), W_E(j)\rangle = -aD_H(i,j) + b$$

Therefore,

$$(W_E)^T W_E = -aD_H + b\mathbf{1}\mathbf{1}^T$$

Let's first look at D_H which has rank at most $m+1$. To see this, let's consider a set of $m+1$ tokens: $\{e_0, e_1, ..., e_m\} \subseteq V$ where $e_k = 2^k$. Here e_0 is associated with the latent vector of all zeroes and the latent vector associated with e_k has only the k-th latent variable being 1.

On the other hand, for any token i, we have that,

$$i = \sum_{k:\iota^{-1}(i)_k=1} e_k$$

In fact,

$$D_H(i) = \sum_{k:\iota^{-1}(i)_k=1} \Big(D_H(e_k) - D_H(e_0) \Big) + D_H(e_0)$$

where $D_H(i)$ is the i-th row of D_H, and for each entry j of $D_H(i)$, we have that

$$D_H(i,j) = \sum_{k:\iota^{-1}(i)_k=1} \Big(D_H(e_k,j) - D_H(e_0,j) \Big) + D_H(e_0,j)$$

This is because

$$D_H(e_k,j) - D_H(e_0,j) = \begin{cases} +1 & \text{if } \iota^{-1}(j)_k = 0 \\ -1 & \text{if } \iota^{-1}(j)_k = 1 \end{cases}$$

Thus, we can rewrite $D_H(i,j)$ as

$$\begin{aligned} D_H(i,j) &= \sum_{k:\iota^{-1}(i)_k=1} \Big(\mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 0] - \mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 1)] \Big) + D_H(e_0,j) \\ &= \sum_{k=1}^{m} \Big(\mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 0] - \mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 1)] \Big) \\ &\qquad + \sum_{k=1}^{m} \Big(\mathbf{1}[\iota^{-1}(i)_k = 0, \iota^{-1}(j)_k = 1] + \mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 1)] \Big) \\ &= \sum_{k=1}^{m} \mathbf{1}[\iota^{-1}(i)_k = 1, \iota^{-1}(j)_k = 0] + \mathbf{1}[\iota^{-1}(i)_k = 0, \iota^{-1}(j)_k = 1] \\ &= D_H(i,j) \end{aligned}$$

Therefore, every row of D_H can be written as a linear combination of $\{D_H(e_0), D_H(e_1), ..., D_H(e_m)\}$.

In other words, D_H has rank at most $m+1$.

Therefore,

$$\mathrm{rank}((W_E)^T W_E) = \mathrm{rank}(W_E) \leq m+2.$$

□

Lemma 39. *Let $z^{(0)}$ and $z^{(1)}$ be two binary vectors of size m where $m \geq 2$. Then,*

$$\sum_{z:D_H(z^{(0)},z)=1} D_H(z, z^{(1)}) = (m-2)D_H(z^{(0)}, z^{(1)}) + m$$

Proof. For z such that $D_H(z, z^{(0)}) = 1$, we know that there are two cases. Either z differs with $z^{(0)}$ on a entry but agrees with $z^{(1)}$ on that entry or z differs with both $z^{(0)}$ and $z^{(1)}$.

For the first case, we know that there are $D_H(z^{(0)}, z^{(1)})$ such entries. In this case, $D_H(z, z^{(1)}) = D_H(z^{(0)}, z^{(1)}) - 1$. For the second case, $D_H(z, z^{(1)}) = D_H(z^{(0)}, z^{(1)}) + 1$.

Therefore,

$$\begin{aligned}
&\sum_{z:D_H(z,z^{(0)})=1} D_H(z, z^{(1)}) \\
&\quad = D_H(z^{(0)}, z^{(1)})(D_H(z^{(0)}, z^{(1)}) - 1) + (m - D_H(z^{(0)}, z^{(1)}))(D_H(z^{(0)}, z^{(1)}) + 1) \\
&\quad = (m-2)D_H(z^{(0)}, z^{(1)}) + m
\end{aligned}$$

□

Lemma 40. *If $m \geq 3$ and $\mathcal{N}(t) = V \setminus \{t\}$, then $\mathcal{N}_1(t) \not\subseteq \mathcal{N}_1(t')$ for any $t, t' \in [V]$.*

Proof. For any token t, $\mathcal{N}_1(t)$ contains any token t' such that $D_H(t, t') = 1$ by the conditions. Then given a set $\mathcal{N}_1(t)$, one can uniquely determine token t. This is because for the set of latent vectors associated with $\mathcal{N}_1(t)$, at each index, there could only be one possible change. □

C.1.4 Proofs for Section 4.4.4

Lemma 29. *Suppose the data generating process follows Section 4.3.1 and $\mathcal{N}(z^*) = \{z : z_1^* = z_1\} \setminus \{z^*\}$. Given the last token in the sequence t_L, then*

$$\nabla_{u_{t,t_L}} \ell(f^L) = \nabla \ell(f^L)^T (W_E)^T W^V (\alpha_t \hat{p}_t W_E(t) - \hat{p}_t \sum_{l=1}^{L} \hat{p}_{t_l} W_E(t_l))$$

where for token t, $\alpha_t = \sum_{l=1}^{L} \mathbf{1}[t_l = t]$ and $\hat{p}_t$ is the normalized attention score for token t.

Proof. Recall that,

$$\begin{aligned} f^L(x) &= \left[{W_E}^T W_V \text{attn}(W_E \chi(x)) \right]_{:L} \\ &= {W_E}^T W_V \sum_{l=1}^{L} \frac{\exp(u_{t_l,t_L})}{Z} W_E(t_l) \end{aligned}$$

where Z is a normalizing constant.

Define $\hat{p}_{t_l} = \frac{\exp(u_{t_l,t_L})}{Z}$. Then we have

$$f^L(x) = {W_E}^T W_V \sum_{l=1}^{L} \hat{p}_{t_l} W_E(t_l)$$

Note that if $t_l = t$ then,

$$\frac{\partial \hat{p}_{t_l}}{\partial u_{t,t_L}} = \hat{p}_{t_l}(1 - \hat{p}_{t_l})$$

Otherwise,

$$\frac{\partial \hat{p}_{t_l}}{\partial u_{t,t_L}} = -\hat{p}_{t_l} \hat{p}_t$$

By the chain rule, we know that

$$\nabla_{u_{t,t_L}} \ell(f^L) = \nabla \ell(f^L)^T (W_E)^T W^V (\sum_{l=1}^{L} \mathbf{1}[t_l = t] \hat{p}_{t_l} W_E(t) - \sum_{l=1}^{L} \hat{p}_{t_l} \hat{p}_t W_E(t_l))$$

Therefore,

$$\nabla_{u_{t,t_L}} \ell(f^L) = \nabla \ell(f^L)^T (W_E)^T W^V (\alpha_t \hat{p}_t W_E(t) - \hat{p}_t \sum_{l=1}^{L} \hat{p}_{t_l} W_E(t_l))$$

where $\alpha_t = \sum_{l=1}^{L} \mathbf{1}[t_l = t]$. □

C.2 Additional experiments – context hijacking

In this section, we show the results of additional context hijacking experiments on the CounterFact dataset [Meng et al., 2022].

Reverse context hijacking In Figure 4.2a, we saw the effects of hijacking by adding in "Do not think of {target_false}." to each context. Now, we measure the effect of the reverse: What if we prepend "Do not think of {target_true}." ?

Based on the study in this work on how associative memory works in LLMs, we should expect the efficacy score to decrease. Indeed, this is what happens, as we see in Figure C.2.1.

Hijacking based on relation IDs We first give an example of each of the 4 relation IDs we hijack in Table C.1.

Table C.1: Examples of contexts in Relation IDs from CounterFact

Relation ID r	Context p	True target o_*	False target $o\_$
P190	Kharkiv is a twin city of	Warsaw	Athens
P103	The native language of Anatole France is	French	English
P641	Hank Aaron professionally plays the sport	baseball	basketball
P131	Kalamazoo County can be found in	Michigan	Indiana

Similar to Figure 4.2b, we repeat the hijacking experiments where we prepend factual sentences generated from the relation ID. We use the format illustrated in Table C.2 for the prepended sentences. We experiment with 3 other relation IDs and we see similar trends

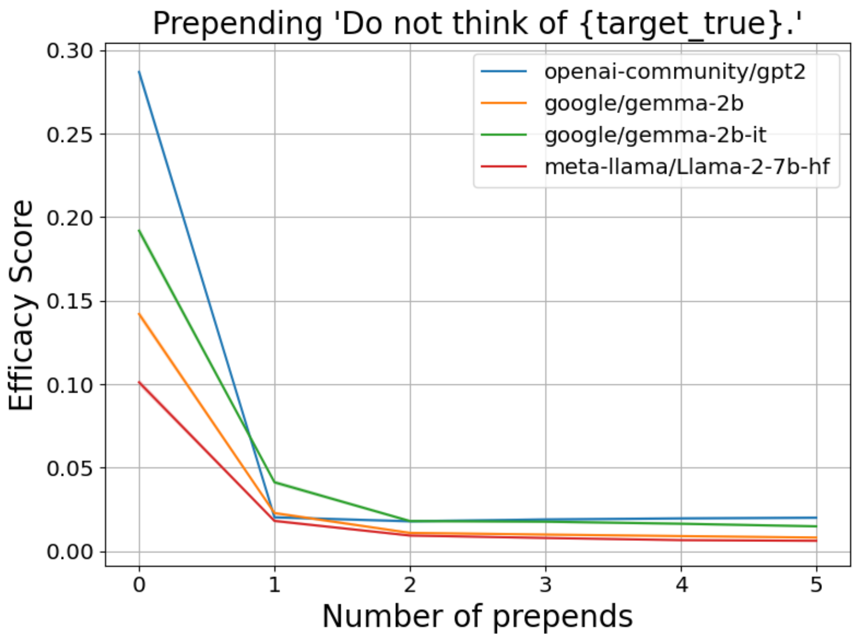


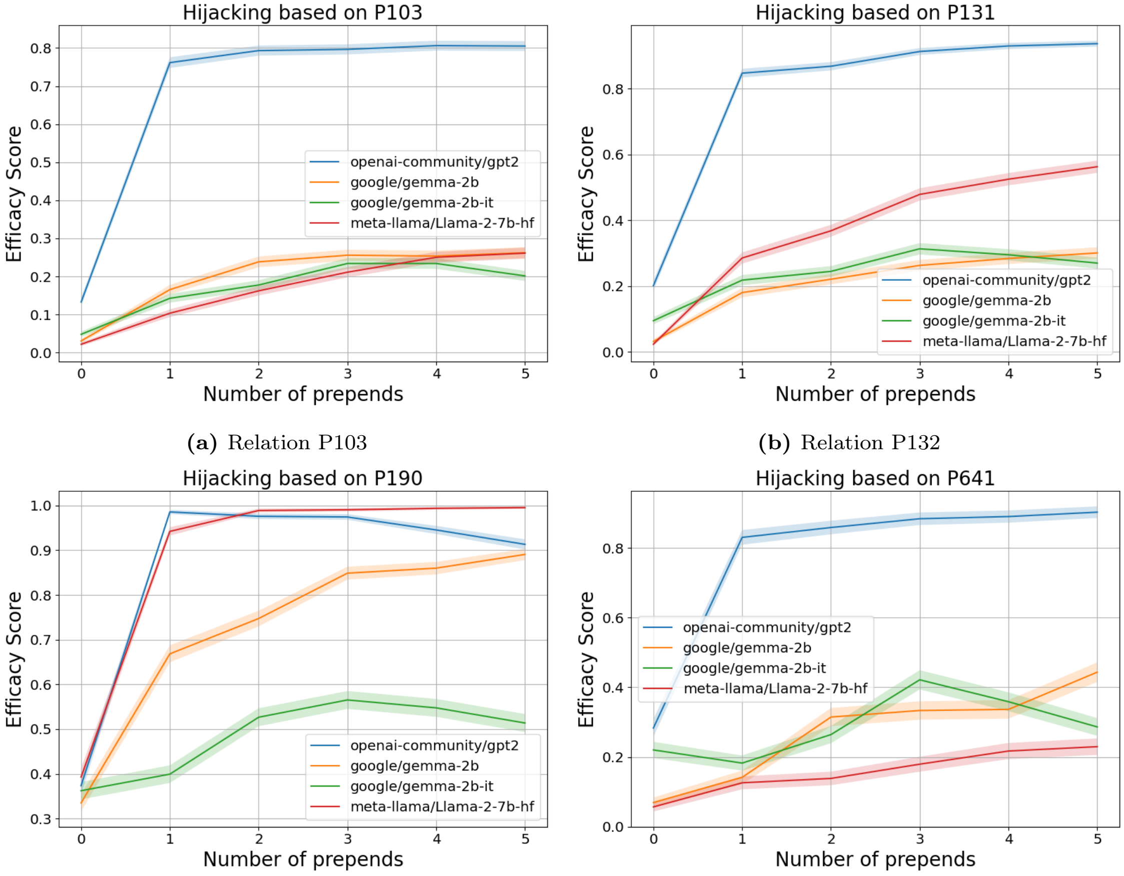
Figure C.2.1: Prepending 'Do not think of {target_true}.' can increase the chance of LLMs to output correct tokens. This figure shows efficacy score versus the number of prepends for various LLMs on the CounterFact dataset with the reverse context hijacking scheme.

Table C.2: Examples of hijack and reverse hijack formats based on Relation IDs

Relation ID r	Context Hijack sentence	Reverse Context Hijack sentence
P190	The twin city of {subject} is not {target_false}	The twin city of {subject} is {target_true}
P103	{subject} cannot speak {target_false}	{subject} can speak {target_true}
P641	{subject} does not play {target_false}	{subject} plays {target_true}
P131	{subject} is not located in {target_false}	{subject} is located in {target_true}

for all the LLMs in Figure C.2.2a, C.2.2b, and C.2.2d. That is, the efficacy score rises for the first prepend and as we increase the number of prepends, the trend of ES rising continues. Therefore, this confirms our intuition that LLMs can be hijacked by contexts without changing the factual meaning.

Similar to Figure C.2.1, we experiment with reverse context hijacking where we give the answers based on relation IDs, as shown in Table C.2. We again experiment with the same 4 relation IDs and the results are in Figure C.2.3a - C.2.3d. We see that the efficacy score decreases when we prepend the answer sentence, thereby verifying the observations of this



(a) Relation P103

(b) Relation P132

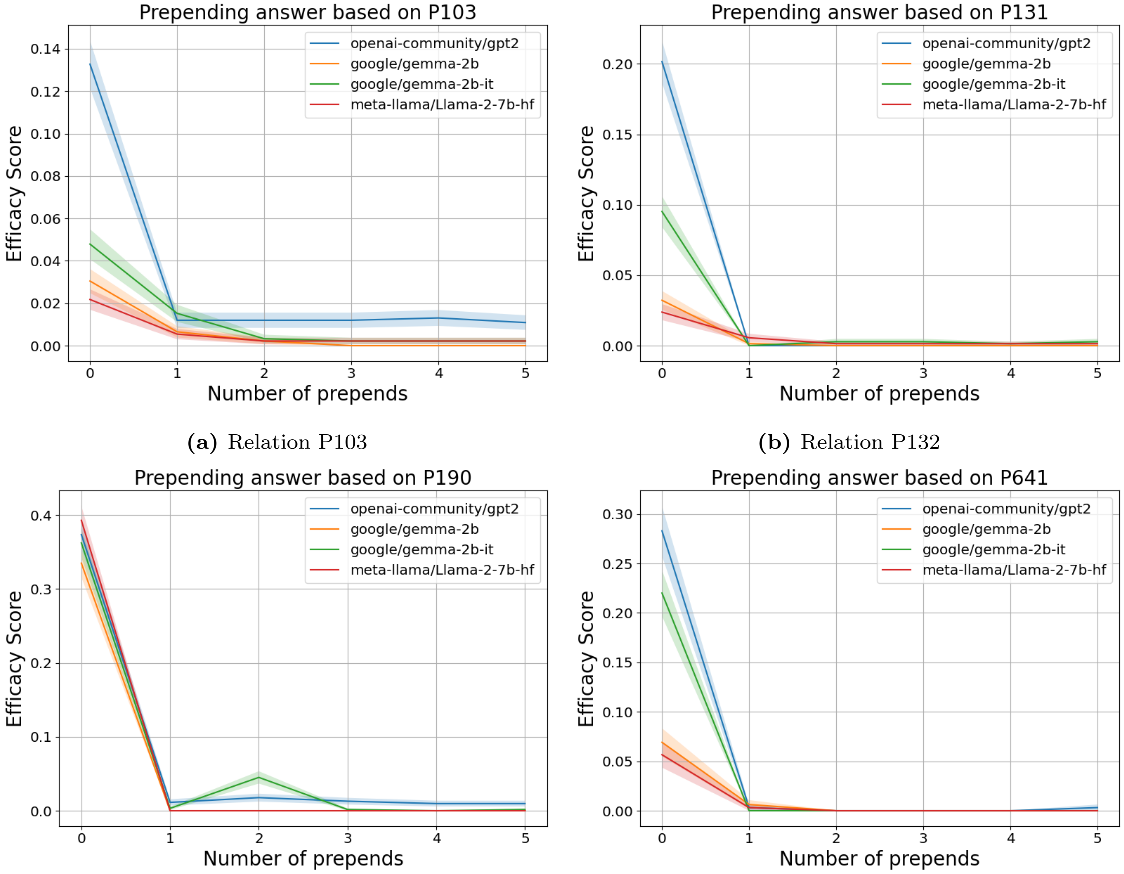
(c) Relation P190

(d) Relation P641

Figure C.2.2: Context hijacking based on relation IDs can result in LLMs output incorrect tokens. This figure shows efficacy score versus the number of prepends for various LLMs on the CounterFact dataset with hijacking scheme presented in Table C.2.

study.

Hijacking without exact target words So far, the experiments use prompts that either contain true or false target words. It turns out, the inclusion of exact target words are not necessary. To see this, we experiment a variant of the generic hijacking and reverse hijacking experiments. But instead of saying "Do not think of {target_false}" or "Do not think of {target_true}". We replace target words with words that are semantically close. Specifically, for relation P1412, we replace words representing language (e.g., "French") with their associated country name (e.g., "France"). As shown in Figure C.2.4, context hijacking



(a) Relation P103

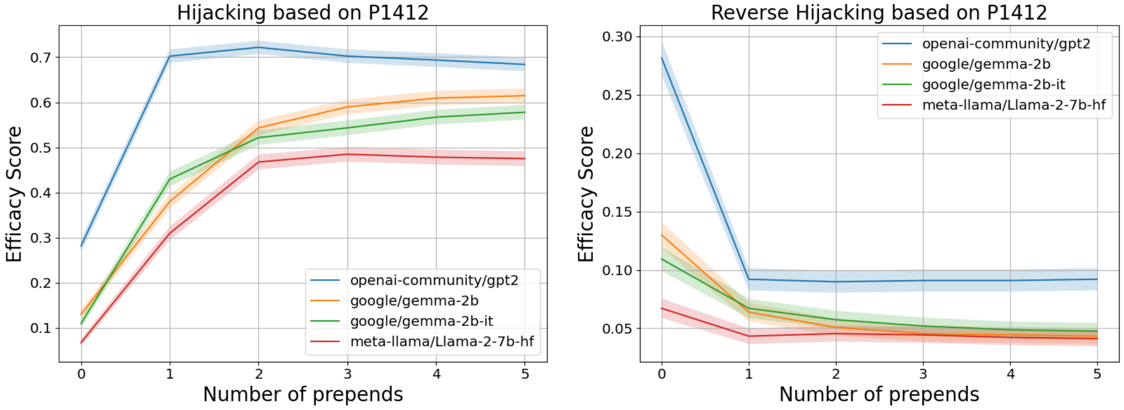
(b) Relation P132

(c) Relation P190

(d) Relation P641

Figure C.2.3: Reverse context hijacking based on relation IDs can result in LLMs to be more likely to be correct. This figure shows efficacy score versus the number of prepends for various LLMs on the CounterFact dataset with the reverse hijacking scheme presented in Table C.2.

and reverse hijacing still work in this case.



(a) Hijacking P1412

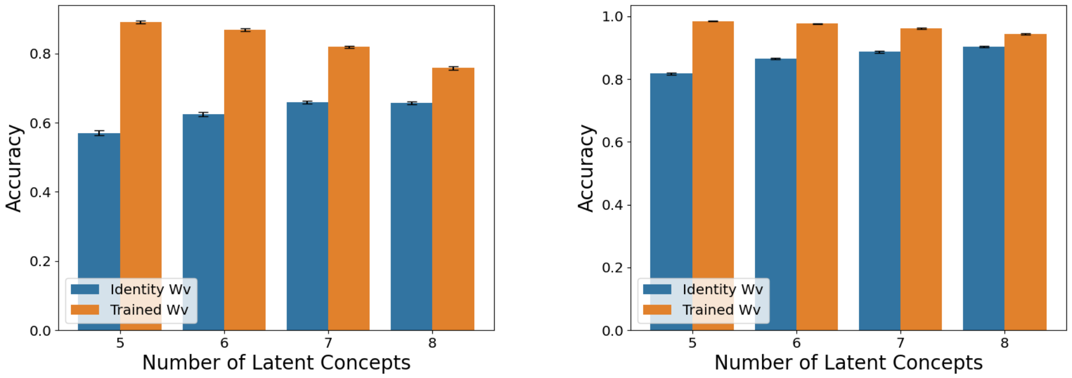
(b) Reverse hijacking P1412

Figure C.2.4: Hijacking and reverse hijacking experiments on relation P1412 show that context hijacking does not require exact target word to appear in the context. This figure shows efficacy score versus the number of prepends for various LLMs on the CounterFact dataset.

C.3 Additional experiments and figures – latent concept association

In this appendix section, we present additional experimental details and results from the synthetic experiments on latent concept association.

Experimental setup Synthetic data are generated following the model in Section 4.3.1. Unless otherwise stated, the default setup has $\omega = 0.5$, $\beta = 1$ and $\mathcal{N}(i) = V \setminus \{i\}$ and $L = 256$. The default hidden dimension of the one-layer transformer is also set to be 256. The model is optimized using AdamW [Loshchilov and Hutter, 2017] where the learning rate is chosen from $\{0.01, 0.001\}$. The evaluation dataset is drawn from the same distribution as the training dataset and consists of 1024 (x, y) pairs. Although theoretical results in Section 4.4 may freeze certain parts of the network for simplicity, in this section, unless otherwise specified, all layers of the transformers are trained jointly. Also, in this section, we typically report accuracy which is $1 -$ error.



(a) $L = 64$

(b) $L = 128$

Figure C.3.1: Fixing the value matrix W_V as the identity matrix results in lower accuracy compared to training W_V, especially for smaller context length L. The figure reports accuracy for both fixed and trained W_V settings, with standard errors calculated over 10 runs.

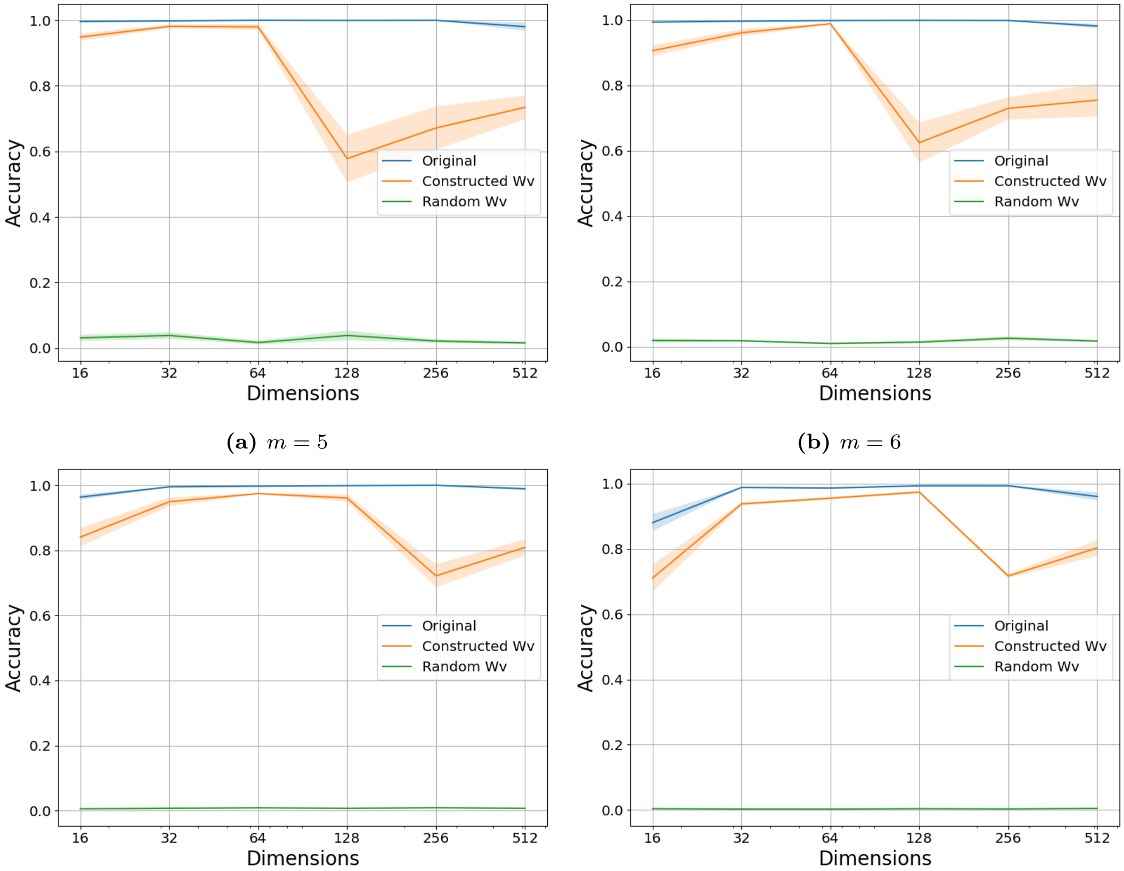
C.3.1 On the value matrix W_V

In this section, we provide additional figures of Section 4.5.1. Specifically, Figure C.3.1 shows that fixing the value matrix to be the identity will negatively impact accuracy. Figure C.3.2 indicates that replacing trained value matrices with constructed ones can preserve accuracy to some extent. Figure C.3.3 suggests that trained value matrices and constructed ones share similar low-rank approximations. For the last two sets of experiments, we consider randomly constructed value matrix, where the outer product pairs are chosen randomly, defined formally as follows:

$$W_V = \sum_{i \in [V]} W_E(i) \Big(\sum_{\{j\} \sim \mathrm{Unif}([V])^{|\mathcal{N}_1(i)|}} W_E(j)^T \Big)$$

C.3.2 On the embeddings

This section provides additional figures from Section 4.5.2. Figure C.3.4 shows that in the underparameterized regime, embedding training is required. Figure C.3.5 indicates that the embedding structure in the underparameterized regime roughly follows (4.4.2). Finally



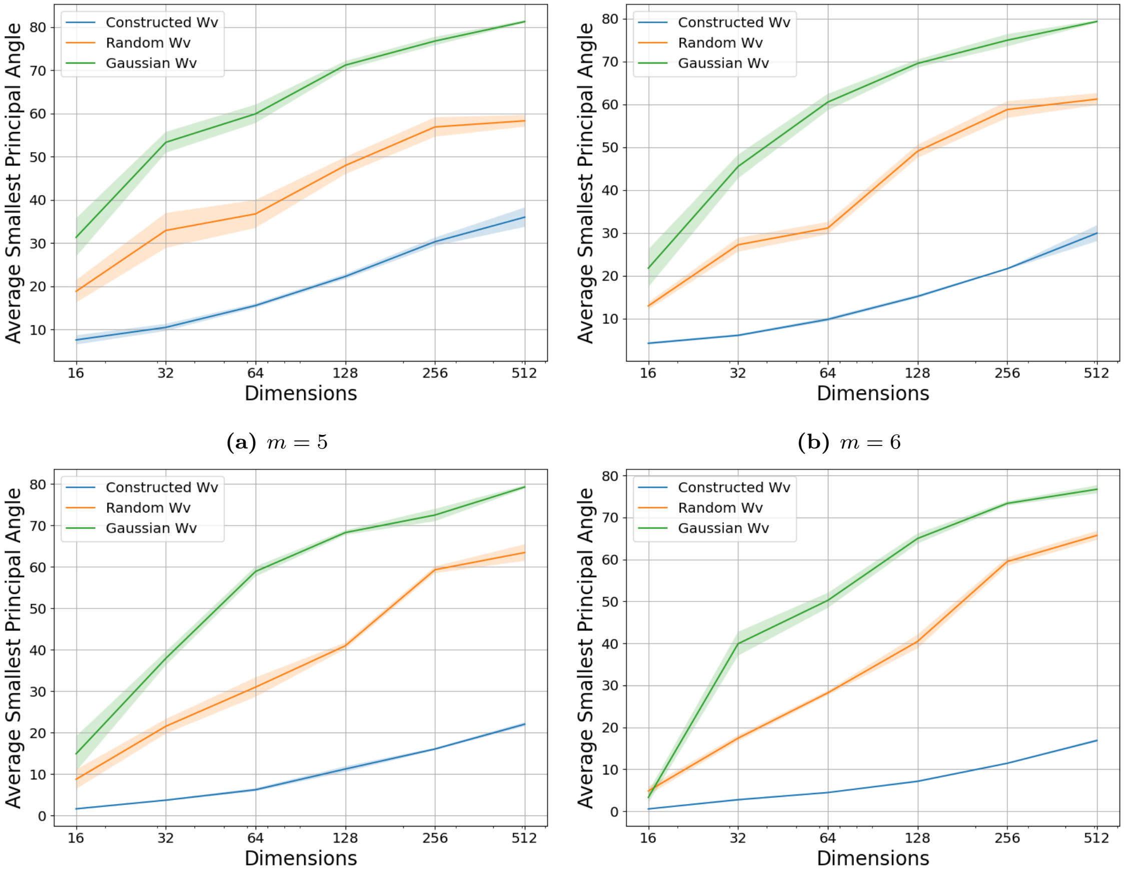
(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.2: When the value matrix is replaced with the constructed one in trained transformers, the accuracy does not significantly decrease compared to replacing the value matrix with randomly constructed ones. The graph reports accuracy under different embedding dimensions and standard errors are over 5 runs.



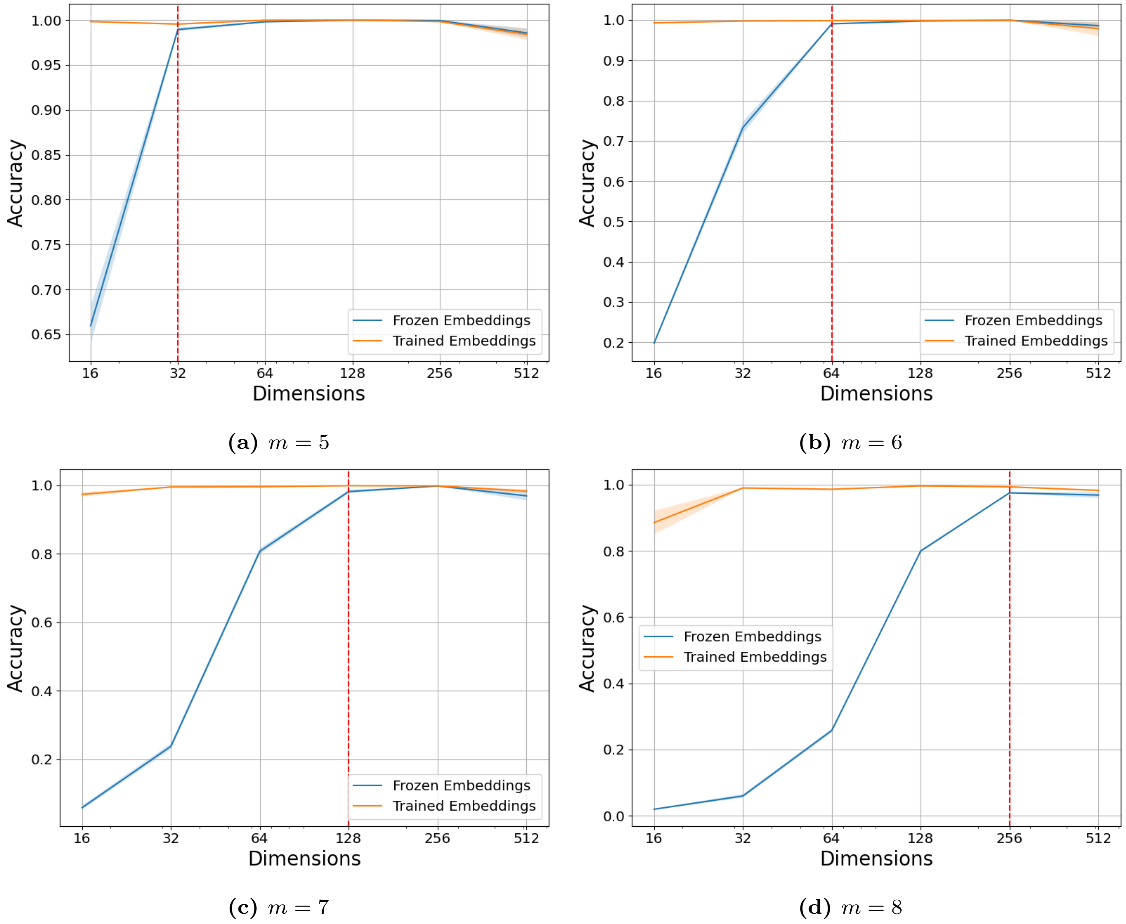
(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.3: The constructed value matrix W_V has similar low rank approximation with the trained value matrix. The figure displays average smallest principal angles between low-rank approximations of trained value matrices and those of constructed, randomly constructed, and Gaussian-initialized value matrices. Standard errors are over 5 runs.



(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

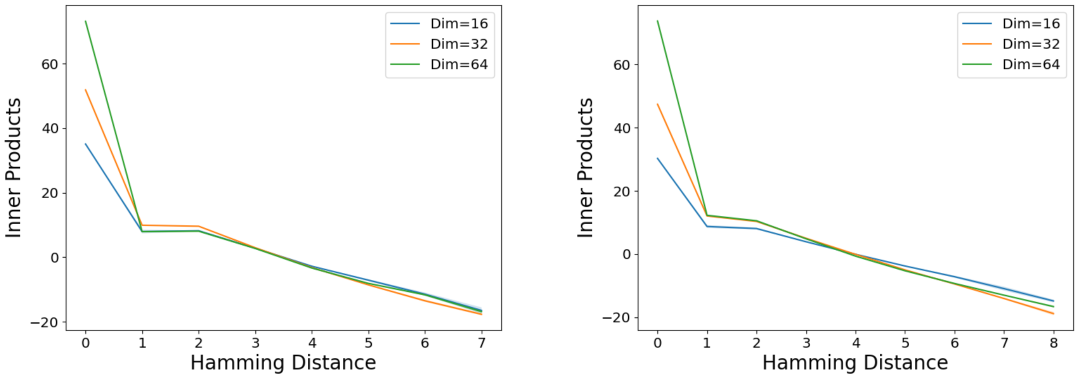
(d) $m = 8$

Figure C.3.4: In the underparameterized regime ($d < V$), freezing embeddings to initializations causes a significant decrease in performance. The graph reports accuracy with different embedding dimensions and the standard errors are over 5 runs. Red lines indicate when $d = V$.

Figure C.3.6 shows that, when the value matrix is fixed to the identity, the relationship between inner product of embeddings and their corresponding Hamming distance is mostly linear.

C.3.3 On the attention selection mechanism

This section provides additional figures from Section 4.5.3. Figure C.3.7-C.3.8 show that attention mechanism selects tokens in the same cluster as the last token. In particular, for Figure C.3.8, we extend experiments to consider cluster structures that depend on the first



(a) $m = 7$ **(b)** $m = 8$

Figure C.3.5: The relationship between inner products of embeddings and corresponding Hamming distances of tokens can be approximated by (4.4.2). The graph displays the average inner product between embeddings of two tokens against the corresponding Hamming distance between these tokens. Standard errors are over 5 runs.

two latent variables. In other words, for any latent vector z^*, we have

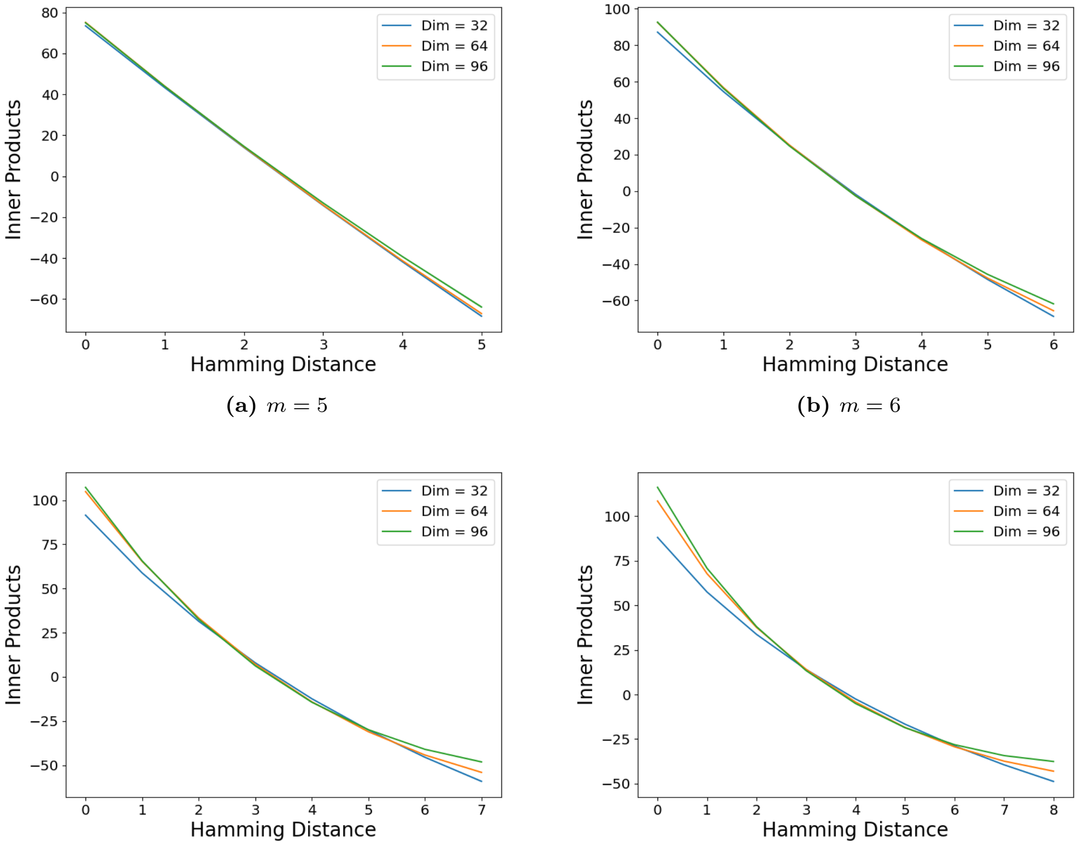
$$\mathcal{N}(z^*) = \{z : z_1^* = z_1 \text{ and } z_2^* = z_2\} \setminus \{z^*\}$$

C.3.4 Spectrum of embeddings

We display several plots of embedding spectra (Figure C.3.9, Figure C.3.10, Figure C.3.11, Figure C.3.12) that exhibit eigengaps between the top and bottom eigenvalues, suggesting low-rank structures.

C.3.5 Context hijacking in latent concept association

In this section, we want to simulate context hijacking in the latent concept association model. To achieve that, we first sample two output tokens y^1 (true target) and y^2 (false target) and then generate contexts $x^1 = (t_1^1, ..., t_L^1)$ and $x^2 = (t_1^2, ..., t_L^2)$ from $p(x^1|y^1)$ and $p(x^2|y^2)$. Then we mix the two contexts with rate p_m. In other words, for the final mixed context $x = (t_1, ..., t_L)$, t_l has probability $1 - p_m$ to be t_l^1 and p_m probability to be t_l^2. Figure C.3.13



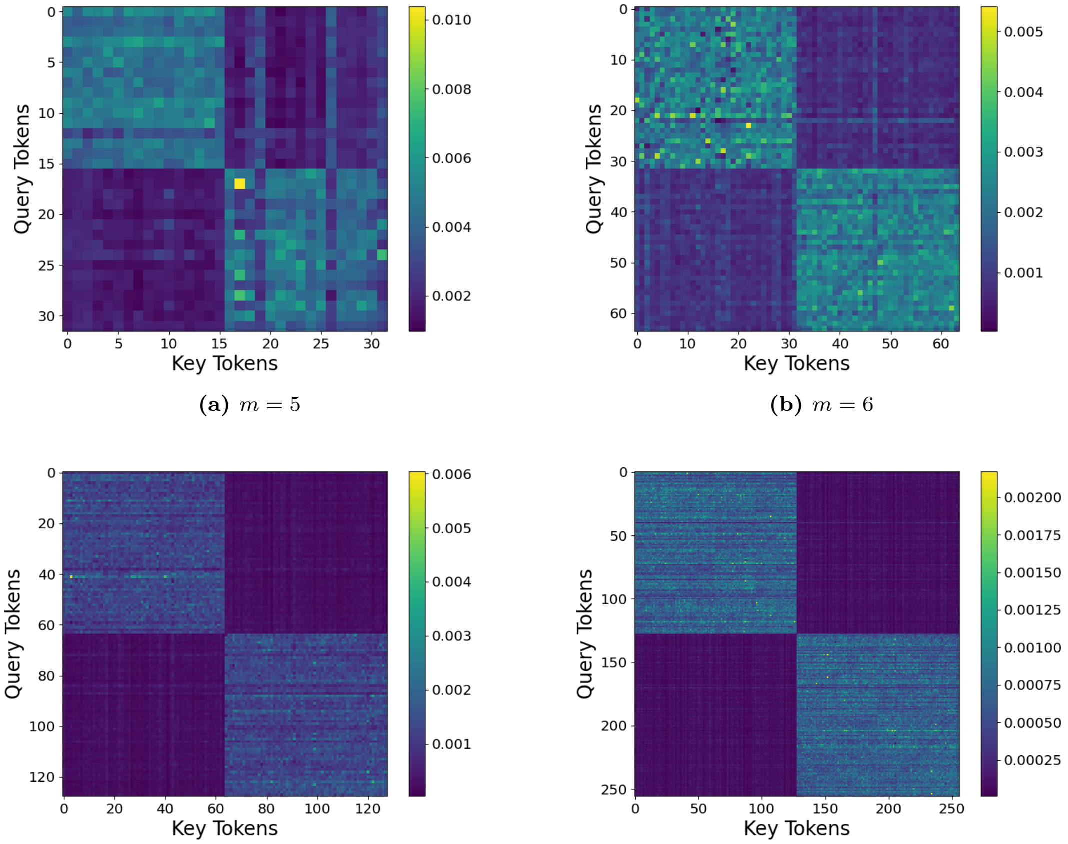
(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.6: The relationship between inner products of embeddings and corresponding Hamming distances of tokens is mostly linear when the value matrix W_V is fixed to be the identity. The graph displays the average inner product between embeddings of two tokens against the corresponding Hamming distance between these tokens. Standard errors are over 10 runs.



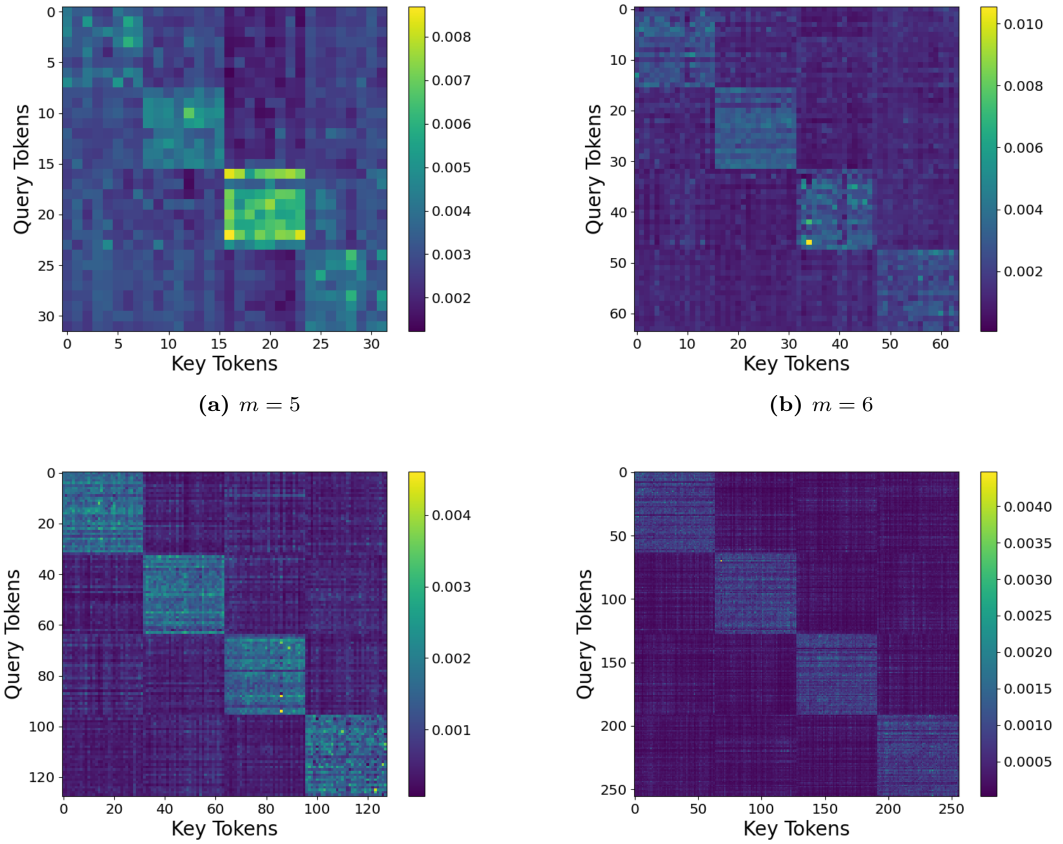
(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.7: The attention patterns show the underlying cluster structure of the data generating process. Here, for any latent vector, we have $\mathcal{N}(z^*) = \{z : z_1^* = z_1\} \setminus \{z^*\}$. The figure shows attention score heat maps that are averaged over 10 runs.



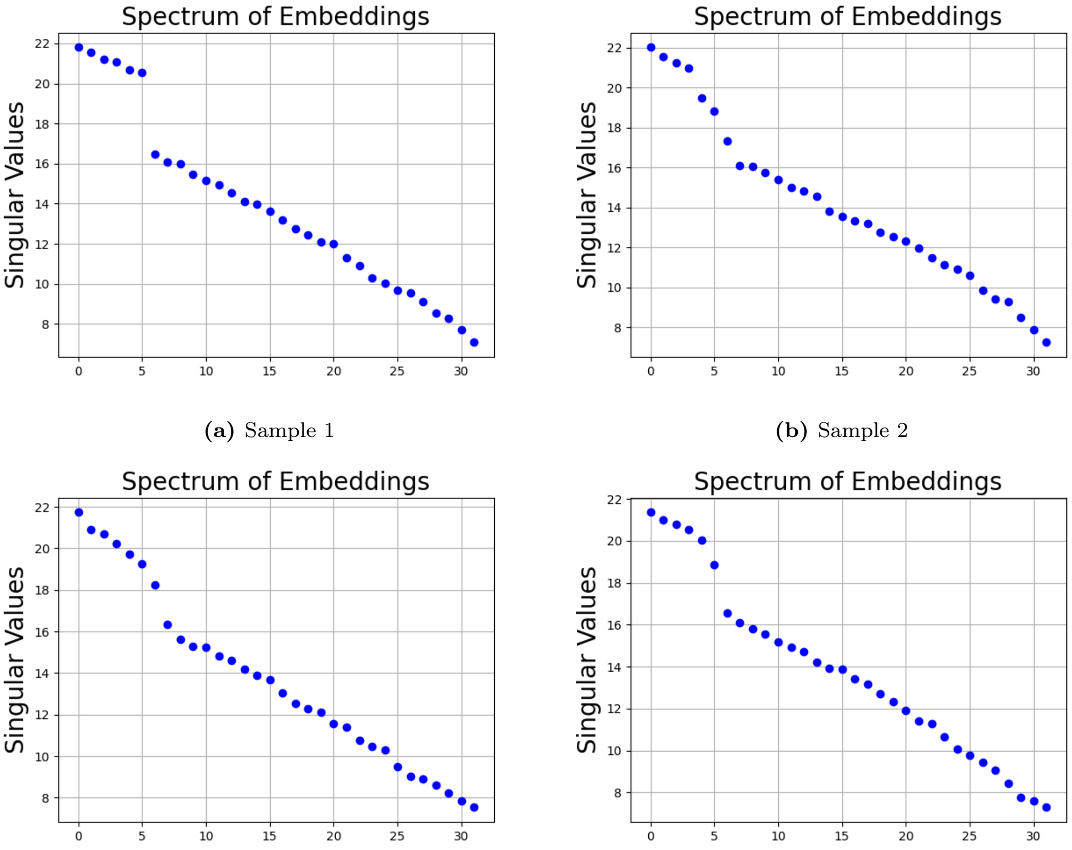
(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.8: The attention patterns show the underlying cluster structure of the data generating process. Here, for any latent vector, we have $\mathcal{N}(z^*) = \{z : z_1^* = z_1 \text{ and } z_2^* = z_2\} \setminus \{z^*\}$. The figure shows attention score heat maps that are averaged over 10 runs.



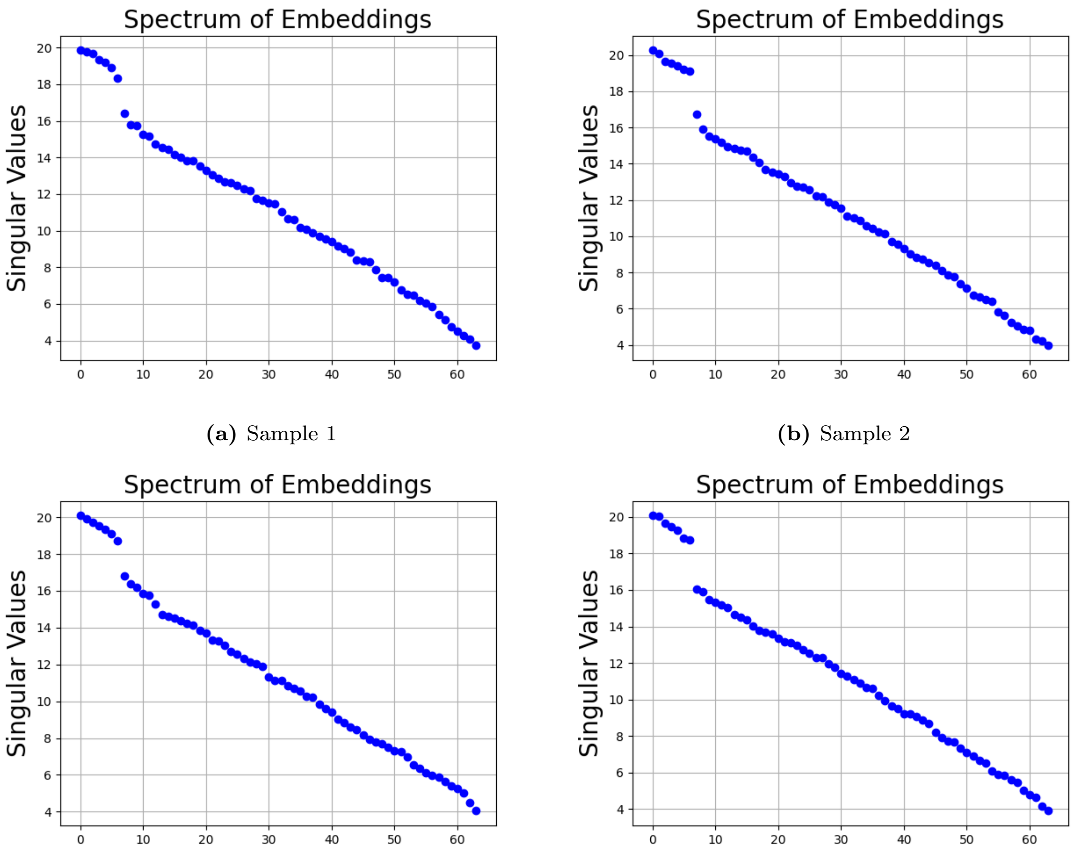
(a) Sample 1

(b) Sample 2

(c) Sample 3

(d) Sample 4

Figure C.3.9: The spectrum of embedding matrix W_E has eigengaps between the top and bottom eigenvalues, indicating low rank structures. The figure shows results from 4 experimental runs. Number of latent variable m is 7 and the embedding dimension is 32.



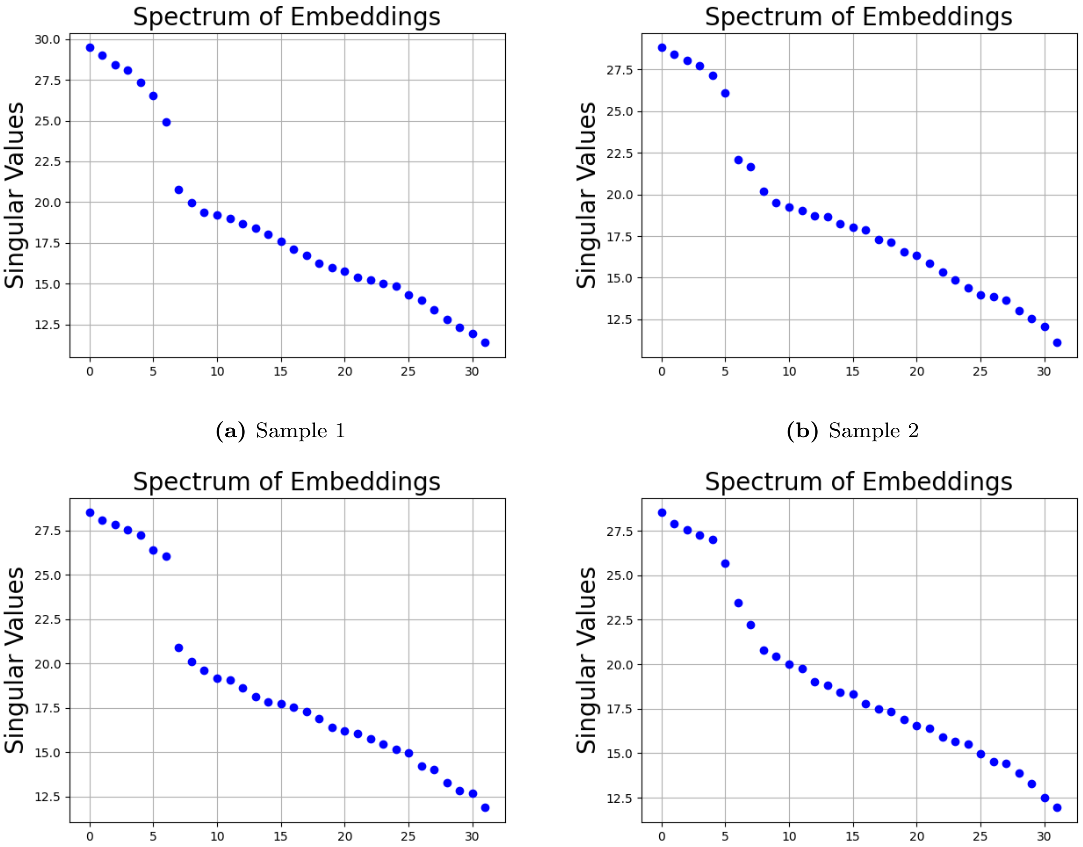
(a) Sample 1

(b) Sample 2

(c) Sample 3

(d) Sample 4

Figure C.3.10: The spectrum of embedding matrix W_E has eigengaps between the top and bottom eigenvalues, indicating low rank structures. The figure shows results from 4 experimental runs. Number of latent variable m is 7 and the embedding dimension is 64.



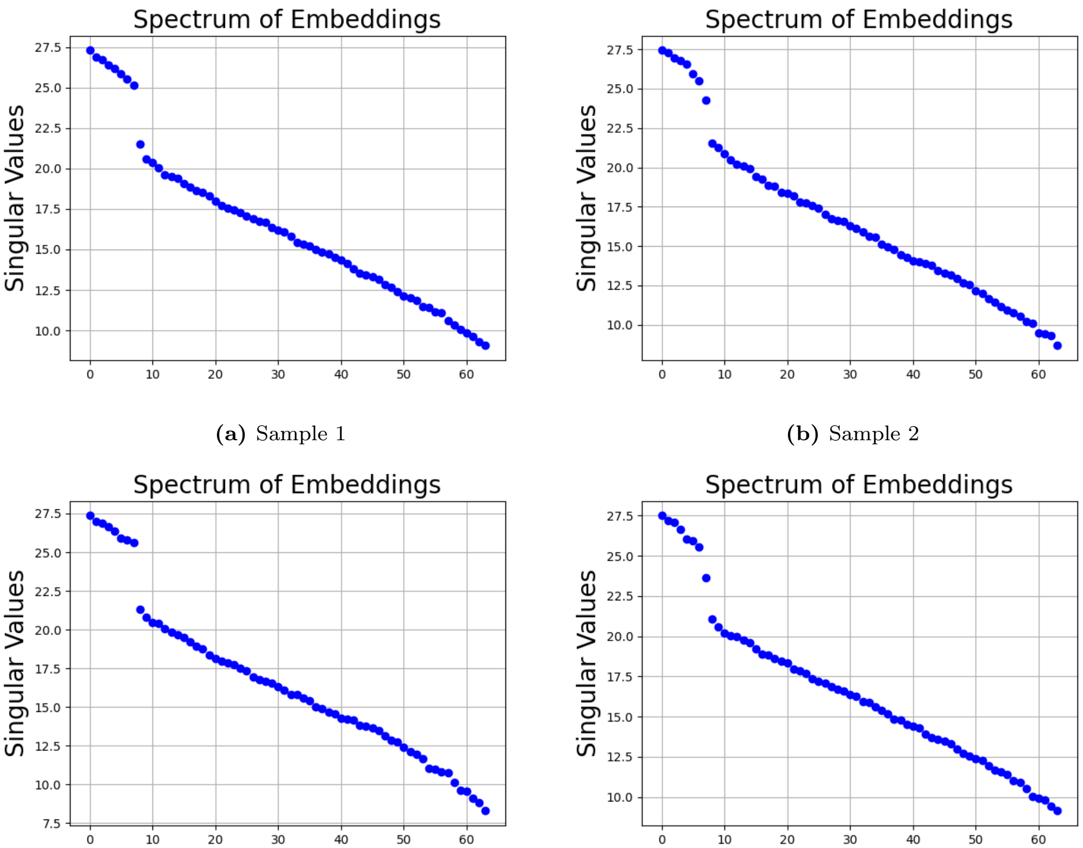
(a) Sample 1

(b) Sample 2

(c) Sample 3

(d) Sample 4

Figure C.3.11: The spectrum of embedding matrix W_E has eigengaps between the top and bottom eigenvalues, indicating low rank structures. The figure shows results from 4 experimental runs. Number of latent variable m is 8 and the embedding dimension is 32.



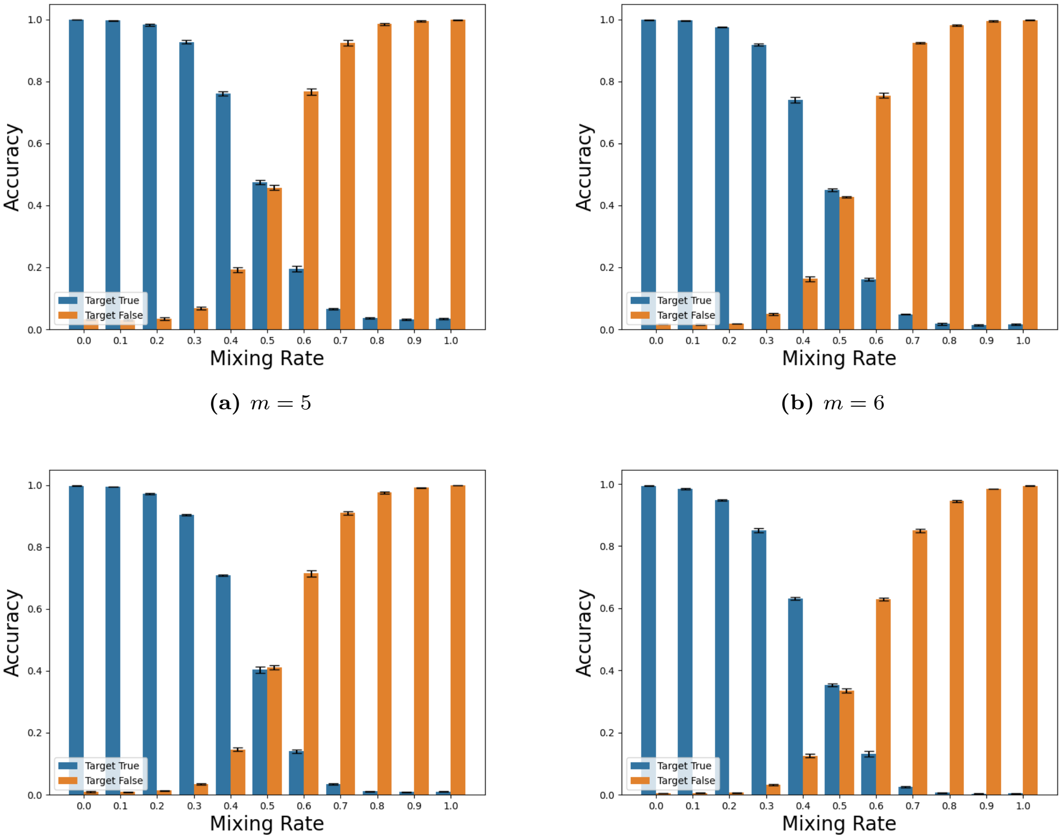
(a) Sample 1

(b) Sample 2

(c) Sample 3

(d) Sample 4

Figure C.3.12: The spectrum of embedding matrix W_E has eigengaps between the top and bottom eigenvalues, indicating low rank structures. The figure shows results from 4 experimental runs. Number of latent variable m is 8 and the embedding dimension is 64.



(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

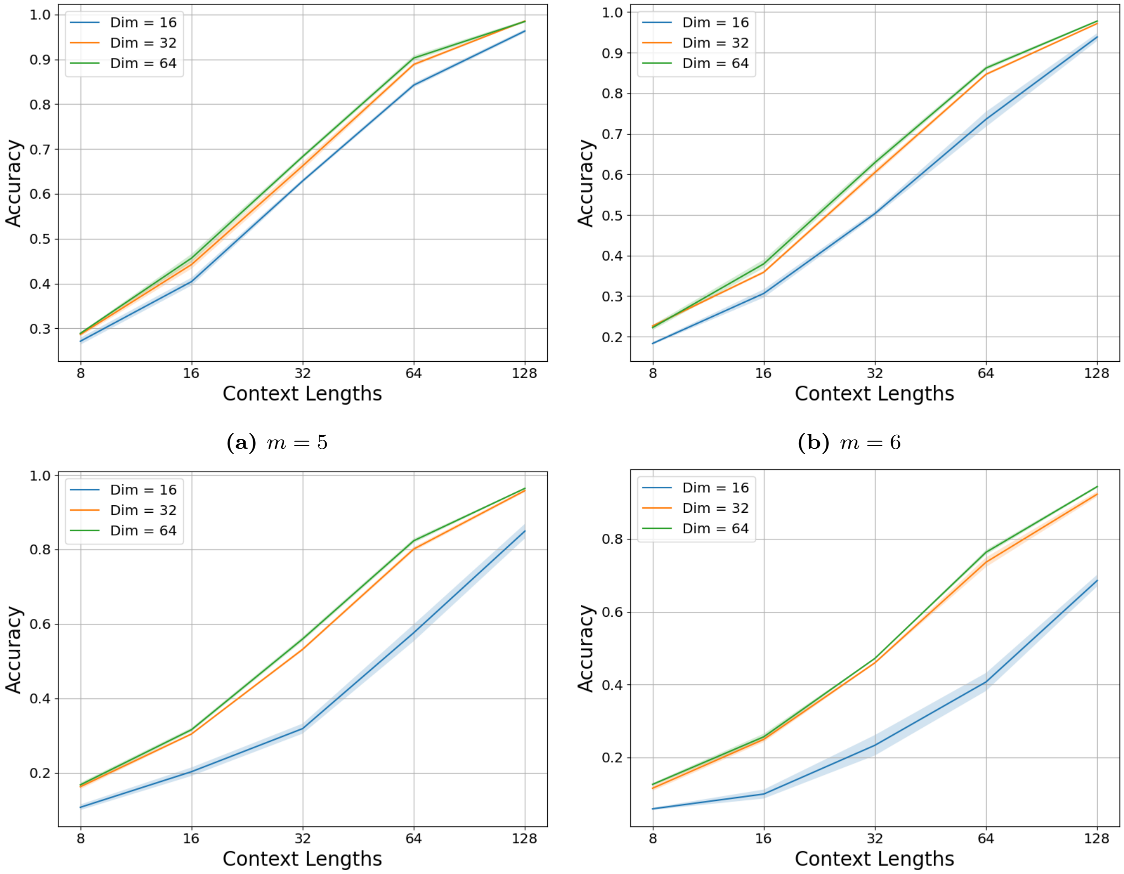
(d) $m = 8$

Figure C.3.13: Mixing contexts can cause misclassification. The figure reports accuracy for true target and false target under various context mixing rate. Standard errors are over 5 runs.

shows that, as the mixing rate increases from 0.0 to 1.0, the trained transformer tends to favor predicting false targets. This mirrors the phenomenon of context hijacking in LLMs.

C.3.6 On the context lengths

As alluded in Section 4.4.5, the memory recall rate is closely related to the KL divergences between context conditional distributions. Because contexts contain mostly i.i.d samples, longer contexts imply larger divergences. This is empirically verified in Figure C.3.14 which demonstrates that longer context lengths can lead to higher accuracy.



(a) $m = 5$

(b) $m = 6$

(c) $m = 7$

(d) $m = 8$

Figure C.3.14: Increasing context lengths can improve accuracy. The figure reports accuracy across various context lengths and dimensions. Standard errors are over 5 runs.