
I Predict Therefore I Am: Is Next Token Prediction Enough to Learn Human-Interpretable Concepts from Data?

Yuhang Liu*

The University of Adelaide

Dong Gong

The University of New South Wales

Yichao Cai

The University of Adelaide

Erdun Gao

The University of Adelaide

Zhen Zhang

The University of Adelaide

Biwei Huang

University of California San Diego

Mingming Gong

The University of Melbourne

Anton van den Hengel

The University of Adelaide

Javen Qinfeng Shi

The University of Adelaide

Abstract

The remarkable achievements of large language models (LLMs) have led many to conclude that they exhibit a form of intelligence. This is as opposed to explanations of their capabilities based on their ability to perform relatively simple manipulations of vast volumes of data. To illuminate the distinction between these explanations, we introduce a novel generative model that generates tokens on the basis of human-interpretable concepts represented as latent discrete variables. Under mild conditions—even when the mapping from the latent space to the observed space is non-invertible—we establish an identifiability result: the representations learned by LLMs through next-token prediction can be approximately modeled as the logarithm of the posterior probabilities of these latent discrete concepts given input context, up to an invertible linear transformation. This theoretical finding not only provides evidence that LLMs capture underlying generative factors, but also provide a unified prospective for understanding of the linear representation hypothesis. Taking this a step further, our finding motivates a reliable evaluation of sparse autoencoders by treating the performance of supervised concept extractors as an upper bound. Pushing this idea even further, it inspires a structural variant that enforces dependence among latent concepts in addition to promoting sparsity. Empirically, we validate our theoretical results through evaluations on both simulation data and the Pythia, Llama, and DeepSeek model families, and demonstrate the effectiveness of our structured sparse autoencoder.

1 Introduction

Large language models (LLMs) are trained on extensive datasets, primarily sourced from the Internet, enabling them to excel in a wide range of downstream tasks, such as language translation, text summarization, and question answering [78, 10, 58]. This remarkable success has ignited a heated debate surrounding the question of whether their performance is primarily achieved through simple

*yuhang.liu01@adelaide.edu.au

operations applied to vast amounts of memorized data, or is a sign of genuine “intelligence”. Intelligence in this context would imply that the LLM had learned a model of the underlying generative factors that gave rise the data.

Recent empirical evidence has made it increasingly clear that the representations learned by LLMs encapsulate latent concepts, such as sentiment [72] or writing style [49], which align with human-interpretable abstractions [1, 50, 65]. Despite this, LLMs are often viewed as mere memorization systems, with their intelligence questioned due to the fundamental simplicity of next-token prediction. Critics argue that this seemingly trivial approach cannot explain the emergence of advanced AI capabilities, much less form the foundation of artificial general intelligence (AGI).

In this context, we thus investigate whether LLMs, despite relying solely on next-token prediction, can grasp human-interpretable concepts, challenging the notion that their success is merely a product of memorization.

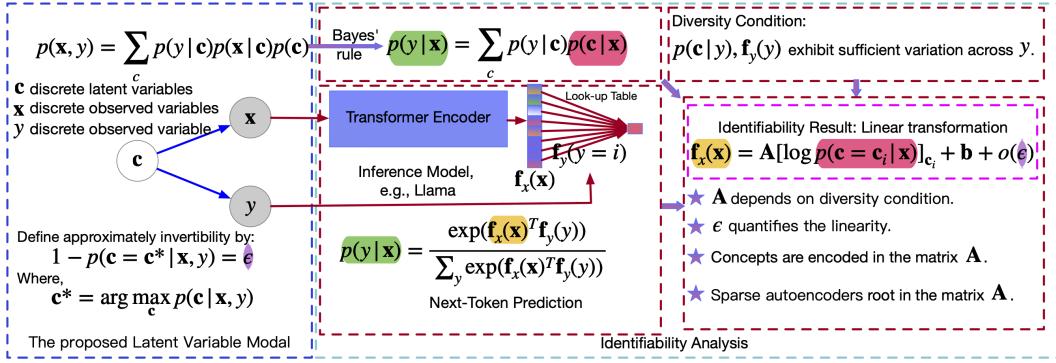


Figure 1: An overview of the main contributions of this work. On the left, we illustrate the proposed latent variable model that represents concepts as latent variables c , which are used to generate both the input x and output y within a next-token prediction framework. Leveraging Bayes’ rule, the next-token prediction framework, and the diversity condition, we establish an identifiability result: the representations learned by LLMs approximately correspond to a linear transformation of the logarithm of the posterior distribution of latent variables conditioned on input tokens, i.e., $f_x(x) = A[\log p(c = c_i|x)]_{c_i} + b + o(\epsilon)$, where b is a constant, and $o(\epsilon)$ represents a term that grows asymptotically smaller than ϵ as $\epsilon \rightarrow 0$. This identifiability result provide a support for the linear representation hypothesis, and sparse autoencoder.

Empirical studies suggest that LLM representations (also known as activations) capture human-interpretable concepts in latent space [72, 1, 50, 65]. To analyze whether and how LLMs learn this through next-token prediction, we introduce a latent variable model, as shown on the left in Figure 1. We model latent concepts as variables connected by arbitrary causal relationships that give rise to observed variables, capturing the underlying generative process between concepts and observations. Although efforts have been taken to model the concepts as continuous linear vectors [63] or explore the empirically recognized linear geometry corresponding to the concepts [59], we aim to develop a model enabling flexibility in capturing the data generative process and theoretical identifiability analysis. We explicitly model all variables, both latent and observed, as discrete, aligning with the inherently discrete nature of language and human-interpretable semantic representations of the world with discreteness. In addition, we do not impose invertibility on the mapping from latent to observed space, as real-world data often involves complex relationships that strict invertibility constraints cannot capture. Building on this model, we develop a theoretical framework to assess whether next-token prediction can learn interpretable concepts. Through identifiability analysis, we show that, under mild conditions, representations learned by next-token prediction approximately correspond to a *linear* transformation of the logarithm of the posterior distribution of latent variables conditioned on input tokens. This connection offers theoretical insights into LLM mechanisms, highlighting their potential to model aspects of human intelligence. Our *linear* identifiability result also offers a unified perspective supporting the *linear* representation hypothesis, which empirically suggests that concepts in LLMs are often encoded linearly [54, 60, 5, 23, 16, 69, 57, 56, 59, 43]. Further, our *linear* identifiability result motivates a reexamination of sparse autoencoders (SAEs), which model LLM representations as sparse *linear* combinations of human-interpretable concepts [32, 13]. We

propose a new evaluation framework for SAEs, treating supervised concept extraction methods, such as linear probing, as an upper bound. More importantly, we introduce a novel SAE variant, structured SAE, that incorporates structured regularization to account for dependencies among latent concepts, in addition to enforcing sparsity. Our main contributions are summarized as follows:

- We introduce a novel latent variable model in Section 2 to formalize the process of language data generation with human-interpretable concepts. This model is designed with relaxed assumptions, aiming to closely approximate real-world language generation.
- In Section 3, we present an identifiability analysis for the proposed latent variable model under next-token prediction framework, showing that the LLM representations learned through it approximate a linear transformation of the logarithm of the posterior of latent concepts in the proposed generative model.
- In Section 4, we show that our identifiability results provide theoretical support and a unified understanding of the linear representation hypothesis in LLMs.
- In Section 5, we introduce a novel approach for evaluating SAEs by treating the performance of supervised concept extractors, e.g., linear probing, as an upper bound. Additionally, we propose structured SAEs, a variant that incorporate structured regularization.
- In Section 6, we present experiments on both simulation data, and real data across the Pythia [8], Llama [70, 18], and DeepSeek-R1 [30] models families, with results consistent with our findings. We also demonstrate the proposed structured SAEs in term of the proposed evaluating method.

The fact that the representation learned by LLMs is equivalent to a causal model of the observed data, up to a linear transformation, shows that next token prediction is enough to support learning a deep understanding of the data. Whether learning the generative model underlying the data constitutes intelligence is a topic for debate, but it is inevitably closer to intelligence than simple manipulations of the training data.

2 A Latent Variable Model for Text Data

We present a novel latent variable model designed to capture the text data generation process, as depicted on the left in Figure 1. The proposed generative model can be expressed as:

$$[\mathbf{x}, y] = \mathbf{g}(\mathbf{c}), \quad \text{or probabilistically, } p(\mathbf{x}, y) = \sum_{\mathbf{c}} p(\mathbf{x}|\mathbf{c})p(y|\mathbf{c})p(\mathbf{c}), \quad (1)$$

where latent variables \mathbf{c} model latent human-interpretable concepts, and observed variables \mathbf{x} and y model words generated by the deterministic mapping \mathbf{g} on \mathbf{c} . We here distinguish observed variables into \mathbf{x} and y to align the input, *i.e.*, context, and output token in the next-token prediction inference framework, respectively.

Our goal is to impose realistic assumptions on the proposed latent variable model, ensuring the model’s flexibility to closely approximate real language data. To this end, we do not impose any specific graph structure over latent variables. For example, we allow for any directed acyclic graph structure over latent variables from the perspective of causal representation learning. Most importantly, we emphasize the following factors.

Discrete Modeling for Language Data. Existing works predominantly assume that both latent and observed variables for modeling language data are continuous [63, 77], primarily to simplify identifiability analysis. However, this assumption overlooks the inherently discrete nature of text. In contrast, we assume throughout this work that all variables, both latent variables \mathbf{c} and observed \mathbf{x} and y , are discrete. This assumption is particularly natural for observed variables \mathbf{x} and y , given the inherently discrete structure of words and categories in text. Likewise, latent variables, which represent underlying concepts, are inherently well-suited to discrete modeling, as they often correspond to categorical distinctions or finite sets of semantic properties that naturally arise in text data. For instance, in a topic modeling scenario, latent variables can represent distinct topics, such as “sports”, “politics”, or “technology”, each of which forms a discrete category. This discrete structure aligns with the way humans typically interpret and classify information in text, making the assumption not only intuitive but also practical for real-world applications [9, 36, 35].

No Invertibility Requirement Unlike many existing approaches [35, 63], we do not require the mapping g from latent to observed variables to be invertible. Allowing non-invertibility is a deliberate design choice driven by the fact that the mapping from latent to observed space is often highly complex and unknown. Imposing constraints on it may unnecessarily limit its flexibility. Moreover, in the context of text data, there are two key considerations. First, in natural language processing, the mapping from latent space to observed space often involves a many-to-one relationship. For example, different combinations of emotional concepts can lead to the same sentiment label. In sentiment analysis, the combination of ‘positive sentiment’ and ‘excitement’ might result in an observed outcome such as ‘This is amazing!’ [55]. Second, some latent concepts may not be explicitly manifested in the observed sentence. For instance, a speaker’s intent, tone, or implicit connotations may influence how a sentence is constructed, yet remain unobservable in the surface-level text. This phenomenon is particularly prominent in pragmatics and discourse analysis, where unspoken contextual factors significantly shape meaning [42]. By not enforcing invertibility, our model can accommodate these unobservable latent factors, which are essential for robust and practical identifiability analysis. To this end, we define the approximate invertibility of the mapping g in Eq. (1) as follows:

Definition 2.1. We define the degree of approximate invertibility of the mapping g in Eq. (1) by introducing an error term ϵ , as follows: $1 - p(\mathbf{c} = \mathbf{c}^* | \mathbf{x}, y) = \epsilon$, where $0 \leq \epsilon < 1$, and $\mathbf{c}^* = \arg \max_{\mathbf{c}} p(\mathbf{c} | \mathbf{x}, y)$ represents the dominant mode of the posterior.

Such relaxed condition naturally implies that we may not achieve exact identifiability results as in previous works, due to inevitable information loss in the context of non-invertible mappings. Nevertheless, this non-invertibility still allows for the consideration of identifiability in an approximate sense, which will be discussed in Section 3.

3 Identifiability Analysis for the Proposed Latent Variable Model

We now turn to the identifiability analysis of the proposed latent variable model, i.e., whether the latent variables \mathbf{c} can be uniquely recovered (up to an equivalence class) from observed data \mathbf{x} and y under additional assumptions. We conduct this analysis within the next-token prediction framework, a widely adopted and empirically significant paradigm for training LLMs [14, 71, 7, 79].

We begin by introducing the general form of next-token prediction, which serves as the foundation for our analysis. The goal of next-token prediction is to learn a model that predicts the conditional distribution over the next token $p(y|\mathbf{x})$, which can be achieved by applying the softmax function over the logits produced by the model’s final layer. This process mirrors multinomial logistic regression, where the model approximates the true conditional distribution $p(y|\mathbf{x})$ by minimizing the cross-entropy loss. In theory, with sufficient training data, a sufficiently expressive architecture, and proper optimization, the model’s predictions will converge to the true conditional distribution. Mathematically,

$$p(y|\mathbf{x}) = \frac{\exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y))}{\sum_{y'} \exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y'))}, \quad (2)$$

where y' denotes a specific value of the observed variable y . The function $\mathbf{f}_x(\mathbf{x})$ maps input \mathbf{x} to a representation space, and $\mathbf{f}_y(y)$ corresponds to the weights in the model’s final layer, i.e., look-up table.

On the other hand, the true $p(y|\mathbf{x})$ can be derived from Eq. (1) using Bayes’ rule:

$$p(y|\mathbf{x}) = \sum_{\mathbf{c}} p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}). \quad (3)$$

By the expression for $p(y|\mathbf{x})$ in both Eq. (2) and Eq. (3), we can align the right-hand sides of these two equations. Taking the logarithm of both sides then yields:

$$\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y) - \log \left(\sum_{y'} \exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y')) \right) = \log \sum_{\mathbf{c}} p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}). \quad (4)$$

We now, as shown in Eq. (4), establish a initial connection between the inference model implemented by LLMs (left-hand side) and the generative model (right-hand side): the left involves representations \mathbf{f}_x learned by LLMs, while the right involves the true posterior $p(\mathbf{c}|\mathbf{x})$.

Diversity Condition To further investigate the relation between the representations \mathbf{f}_x and the true posterior $p(\mathbf{c}|\mathbf{x})$ in Eq. (4), we introduce the following assumption, referred to as the *diversity condition*. Formally, we assume that: there exist $\ell+1$ distinct values of y , i.e., $y_0, \dots, y_k, \dots, y_\ell$, such that the matrix $\mathbf{L} = ([p(\mathbf{c} = \mathbf{c}_i|y = y_1) - p(\mathbf{c} = \mathbf{c}_i|y = y_0)]_{\mathbf{c}_i}, \dots, [p(\mathbf{c} = \mathbf{c}_i|y = y_\ell) - p(\mathbf{c} = \mathbf{c}_i|y = y_0)]_{\mathbf{c}_i})$ of size $\ell \times \ell$ is invertible, where ℓ is the number of all possible values that \mathbf{c} can take², the notation $[\cdot]_{\cdot}$, throughout the paper, denotes the vector indexed by possible values of latent variables. When referring to the full latent variables \mathbf{c} , we write $[\cdot]_{\mathbf{c}_i}$, when referring to a single concept c^i , we write $[\cdot]_{c^i}$. This assumption was initially developed in the context of nonlinear independent component analysis [33, 34, 39]. Basically, it requires that the distribution $p(\mathbf{c}|y)$ exhibits sufficiently strong and diverse variation across different values of y . Similarly, we also assume that there exist $\ell+1$ values of y , so that the matrix $\hat{\mathbf{L}} = (\mathbf{f}_y(y = y_1) - \mathbf{f}_y(y = y_0), \dots, \mathbf{f}_y(y = y_\ell) - \mathbf{f}_y(y = y_0))$ of size $\ell \times \ell$ is also invertible. This assumption has been employed in the context of *identifiability analysis in inference space* for LLMs [64, 51]. Under this diversity condition, we derive the following identifiability result:

Theorem 3.1. *Under the diversity condition above, the true latent variables \mathbf{c} are related to the representations in LLMs, i.e., $\mathbf{f}_x(\mathbf{x})$, which are learned through the next-token prediction framework, by the following relationship:*

$$\mathbf{f}_x(\mathbf{x}) = \mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i} + \mathbf{b} - (\hat{\mathbf{L}}^T)^{-1}\mathbf{h}_y, \quad (5)$$

where $\mathbf{h}_y = [h_{y_1} - h_{y_0}, \dots, h_{y_\ell} - h_{y_0}]$ with $h_{y_k} = [p(\mathbf{c} = \mathbf{c}_i|y = y_k)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i|y = y_k, \mathbf{x})]_{\mathbf{c}_i}$, and $\mathbf{b} = (\hat{\mathbf{L}}^T)^{-1}\mathbf{b}_y$, with $\mathbf{b}_y = [b(y = y_1) - b(y = y_0), \dots, b(y = y_\ell) - b(y = y_0)]$ and $b(y = y_k) = \mathbb{E}_{p(\mathbf{c}|y=y_k)}[\log p(y = y_k|\mathbf{c})]$, and $\mathbf{A} = (\hat{\mathbf{L}}^T)^{-1}\mathbf{L}$. When $\epsilon = 0$, $\mathbf{f}_x(\mathbf{x}) = \mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i} + \mathbf{b}$. As $\epsilon \rightarrow 0$, $\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i} + \mathbf{b}$.

Discussion Identifiability analysis is a fundamental challenge, especially in the causal representation learning community. While this community often assumes that the mapping from latent space to observed space is invertible [12, 74, 53, 75, 3, 67, 68, 44], such an assumption may, in some real-world applications, oversimplify the inherently complex relationships between latent and observed variables. For example, in financial data, different combinations of latent factors (e.g., interest rates, geopolitical events) may lead to the same outcomes, making the mapping non-invertible. In healthcare, varied genetic and environmental factors may produce identical disease outcomes. These examples illustrate the non-invertibility of such mappings. We hope this work will inspire further research aimed at overcoming the limitations of invertibility assumptions in causal representation learning.

4 Understanding and Unifying the Linear Representation Hypothesis

In this section, we demonstrate how the identifiability result presented in Theorem 3.1 supports the linear representation hypothesis in LLMs. To achieve this, we first briefly introduce the linear representation hypothesis and then explain how it can be understood and unified through the linear matrix \mathbf{A} from our identifiability result.

4.1 The Linear Representation Hypothesis

The linear representation hypothesis suggests that human-interpretable concepts in LLMs are represented linearly. This idea is supported by empirical evidence in various forms, including:

Concepts as Directions: Each concept is represented as a direction determined by the differences (i.e., vector offset) in representations of pairs that vary only in one latent concept of interest, e.g., gender. For instance, $\text{Rep}(\text{'men'}) - \text{Rep}(\text{'women'}) \approx \text{Rep}(\text{'king'}) - \text{Rep}(\text{'queen'})$ [54, 60, 72].

Concept Manipulability: Previous studies have demonstrated that the value of a concept can be altered independently from others by introducing a corresponding steering vector [43, 76, 72]. For example, transitioning an output from a false to a truthful answer can be accomplished by adding a vector offset derived from counterfactual pairs that differ solely in the false/truthful concept. This illustrates how such vectors encapsulate the essence of false/truthful concepts.

²For example, if $\mathbf{c} = [c^1, c^2]$, and $c^1 \in \{0, 1\}, c^2 \in \{0, 1, 2\}$, then \mathbf{c} can take 2×3 values.

Linear Probing: The value of a concept is often measured using a linear probe. For instance, the probability that the output language is French is logit-linear in the representation of the input. In this context, the linear weights can be interpreted as representing the concept of English/French [59, 52].

4.2 Understanding the Linear Representation Hypothesis

The linear representation hypothesis has received growing empirical support in recent years. While recent work has aimed to develop unified frameworks for a deeper understanding of this phenomenon [59, 51, 35], our approach seeks to explain it through the lens of identifiability. Before that, we first provide the following definition:

Definition 4.1. We define the degree of approximate invertibility of the mapping from \mathbf{c} to \mathbf{x} by introducing an error term $\epsilon_{\mathbf{x}}$, as follows: $1 - p(\mathbf{c} = \mathbf{c}^* | \mathbf{x}) = \epsilon_{\mathbf{x}}$, where $0 < \epsilon_{\mathbf{x}} < 1$, and $\mathbf{c}^* = \arg \max_{\mathbf{c}} p(\mathbf{c} | \mathbf{x})$ represents the dominant mode of $p(\mathbf{c} | \mathbf{x})$.

We then present two key corollaries derived from our identifiability results.

Corollary 4.2 (Concepts Are Encoded in the Matrix \mathbf{A}). *Suppose that Theorem 3.1 holds, i.e., $\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i} + \mathbf{b}$. Let \mathbf{x}_0 and \mathbf{x}_1 be a pair of inputs that differ only in the i -th concept variable c^i . Then when $\epsilon_{\mathbf{x}} \rightarrow 0$ in Def. 4.1, $\mathbf{f}_x(\mathbf{x}_1) - \mathbf{f}_x(\mathbf{x}_0) \approx \tilde{\mathbf{A}}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c_i})$, where $\tilde{\mathbf{A}}^i = \mathbf{AB}^i$, where \mathbf{B}^i is a binary lifting matrix that broadcasts each entry of $[\log p(c^i | \mathbf{x})]_{c_i}$ to the corresponding index in $[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i}$.*

Understanding Concepts as Directions The corollary explains why the representation difference $\text{Rep}(\text{'man'}) - \text{Rep}(\text{'woman'})$ can be closely approximated by $\text{Rep}(\text{'king'}) - \text{Rep}(\text{'queen'})$. In both the ('man', 'woman') and ('king', 'queen') pairs, the primary distinguishing factor is the latent concept of gender, while other concepts such as royalty remain largely unchanged. As a result, both $\text{Rep}(\text{'man'}) - \text{Rep}(\text{'woman'})$ and $\text{Rep}(\text{'king'}) - \text{Rep}(\text{'queen'})$ are driven by the same expression $\tilde{\mathbf{A}}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c_i})$. In the case of a binary concept c^i , this expression defines a directional vector corresponding to the concept of interest. See Appendix F.1 for details. .

Understanding Concept Manipulability The corollary also supports the notion that a concept's value can be adjusted by adding a corresponding steering vector, such as $\text{Rep}(\text{'man'}) - \text{Rep}(\text{'woman'})$. Adding the steering vector effectively modifies the original representation to produce a new representation, i.e., $\hat{\mathbf{f}}_{\mathbf{x}} = \mathbf{f}_{\mathbf{x}}(\mathbf{x}) + \alpha (\tilde{\mathbf{A}}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c_i})) = \mathbf{A} ([\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i} + \alpha \mathbf{B}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c_i})) + \mathbf{b}$, where α corresponds to an introduced weight [76, 72]. Thus, manipulating a concept via a steering vector is, in essence, equivalent to modifying the posterior distribution of the concept of interest, which directly impacts the model's output.

To understand **Linear Probing**, we first introduce the following corollary:

Corollary 4.3 (Linear Classifiability of Representations). *Suppose that Theorem 3.1 holds, i.e., $\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i} + \mathbf{b}$. Let \mathbf{x}_0 and \mathbf{x}_1 be pair data that differ only in the i -th concept variable c^i , with labels c^i . Then when $\epsilon_{\mathbf{x}} \rightarrow 0$ in Def. 4.1, the corresponding representations $(\mathbf{f}(\mathbf{x}_0), \mathbf{f}(\mathbf{x}_1))$ are linearly separable with a weight matrix \mathbf{W} satisfying $\mathbf{W}\tilde{\mathbf{A}}^i \approx s\mathbf{I}$, where s accounts for softmax scaling. The corresponding logit is the unnormalized $[p(c^i | \mathbf{x})]_{c_i}$.*

This corollary supports that latent concepts, e.g., English vs. French, can be reliably predicted using a linear probe on the model's internal representations. This linear separability enables alignment between the model's predictive distribution and the true class distribution, achieved via cross-entropy minimization. A specific analysis for the binary case of concept c^i is provided in Appendix F.2.

4.3 Unifying the Linear Representation Hypothesis

Corollary 4.3 reveals that the weights \mathbf{W} learned through linear probing are intimately connected to the linear transformation \mathbf{A} from our identifiability results, with the key relationship $\mathbf{W}\tilde{\mathbf{A}}^i \approx s\mathbf{I}$. This relationship, previously unexplored, provides a unified framework for understanding the linear representation hypothesis across various forms, e.g., concepts as directions, concept manipulability, and linear probing. By leveraging the relationship $\mathbf{W}\tilde{\mathbf{A}}^i \approx s\mathbf{I}$, we demonstrate that these different forms of the linear representation hypothesis are not independent, but are instead intrinsically linked

through the same linear transformation \mathbf{A} from our identifiability result. This unified viewpoint underscores how \mathbf{A} serves as the underlying structure that connects concept as directions, concept manipulation, and linear probing.

5 Evaluating Sparse Autoencoders and Structured Sparse Autoencoders

Evaluating SAEs Broadly speaking, SAEs are designed with two primary objectives. First, they aim to learn a set of latent features \mathbf{z} such that sparse linear combinations $\beta\mathbf{z}$ can accurately reconstruct internal representations from LLMs, i.e., $\mathbf{f}(\mathbf{x}) \approx \beta\mathbf{z}$. Second, they seek to ensure that each learned feature z_i corresponds to a disentangled, human-interpretable concept, thereby enabling a mechanistic understanding of LLMs. While reconstruction loss is commonly used to assess how well representations are reconstructed [61, 62, 27, 11], it is a limited proxy for the second objective. A key challenge lies in the absence of ground truth for the underlying concepts [37], making the evaluation of feature disentanglement particularly difficult.

To address it, we propose a new evaluation method for SAEs grounded in our theoretical insights. Specifically, based on Theorem 3.1, the LLM representations $\mathbf{f}_x(\mathbf{x})$ can be approximated by $\mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i}$. Meanwhile, SAEs are trained to reconstruct the same $\mathbf{f}_x(\mathbf{x})$ by $\beta\mathbf{z}$. Combining the two, we arrive at: $\beta\mathbf{z} \approx \mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i}$. This suggests that SAE features \mathbf{z} are linearly related to $[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i}$. Based on this, if we expect that each z_i encodes a single concept c^i , then it naturally follows that z_i should be linearly related only to the posterior of a single concept $\log p(c^i | \mathbf{x})$. Consequently, we can evaluate whether each z_i has successfully learned a disentangled concept by measuring its linear correlation with $\log p(c^i | \mathbf{x})$. The question is how can we obtain $p(c^i | \mathbf{x})$.

Based on Corollary 4.3, we can obtain $p(c^i | \mathbf{x})$ by pair data that differ only in the i -th concept variable c^i . Specifically, we can construct paired data $(\mathbf{x}_0, \mathbf{x}_1)$ that differ in only a single *binary* concept c^i , with labels $c^i = 0$ for \mathbf{x}_0 and $c^i = 1$ for \mathbf{x}_1 . We then train a linear classifier (i.e., linear probing) in a supervised manner on the corresponding LLM representations $\mathbf{f}(\mathbf{x}_0)$ and $\mathbf{f}(\mathbf{x}_1)$, with the goal of predicting c^i . Once trained, the corresponding logit provides a reliable estimate of the posterior probability $p(c^i = 1 | \mathbf{x})$. See Appendix F.2 for a binary special case of Corollary 4.3.

Structured SAEs Again, building on $\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A}[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i}$ in Theorem 3.1, SAEs attempt to recovering interpretable concepts from entangled representations in a unsupervised manner, which is typically an ill-posed inverse problem. As a result, using sparsity-inducing prior only, e.g., ℓ_p [38] or top- k regularization [26, 17], may be insufficient for regularizing the solution space. To address this limitation, by recognising concepts may appear dependencies, we propose structured SAEs, which incorporate additional constraints to better reflect the inductive biases of the latent concept space by accounting for potential dependencies among concepts. In particular, we introduce *low-rank regularization* to complement sparsity-based priors (Other forms of structured regularization may also be worth exploring.). Formally, structured SAEs minimize the following objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{train}}} [\|\mathbf{f}(\mathbf{x}) - \bar{\mathbf{f}}(\mathbf{x})\|_2^2 + \lambda_t (\|\mathbf{S}\|_{p_t}^{p_t} + \gamma \|\mathbf{R}\|_{\text{nuc}})], \quad (6)$$

where $\bar{\mathbf{f}}(\mathbf{x})$ denotes reconstruction, λ_t is the dynamically adjusted sparsity coefficient at step t , following the p -annealing SAE strategy [38], γ is a hyperparameter that balances the sparsity penalty and the low-rank regularization, $\|\mathbf{S}\|_{p_t}^{p_t} = \sum_i |S_i|^{p_t}$ is the adaptive ℓ_{p_t} norm promoting sparsity, $\|\mathbf{R}\|_{\text{nuc}}$ is the nuclear norm, used to encourage low-rank structure on \mathbf{R} , and $\mathbf{z} = \mathbf{S} + \mathbf{R}$.

6 Empirical Evaluation on Simulated and Real Data with LLMs

Simulation We begin by conducting experiments on synthetic data, which is generated through the following process: First, we create random directed acyclic graphs (DAGs) with n latent variables, representing concepts. For each random DAG, the conditional probabilities of each variable given its parents are modeled using Bernoulli distributions, where the parameters are sampled uniformly from $[0.2, 0.8]$. To simulate a nonlinear mixture process, we then convert the latent variable samples into one-hot format and randomly apply a permutation matrix to the one-hot encoding, generating one-hot observed samples. These are then transformed into binary observed samples. To simulate next-token prediction, we randomly mask part of the binary observed data, i.e., x_i , and use the remaining portion, i.e., \mathbf{x}_j where $j \neq i$, to predict the masked part x_i . Refer to Section G.1 in Appendix for more details.

Evaluation In Theorem 3.1, we demonstrate that the features learned through next-token prediction approximate a linear transformation of $\log p(\mathbf{c}|\mathbf{x})$. This approximation becomes tighter when the mapping from \mathbf{c} to \mathbf{x} is approximately invertible, as analyzed in Theorem 3.1. Building on this, Corollary 4.3 establishes that for a data pair (\mathbf{c}, \mathbf{x}) , the representations $f_{\mathbf{x}}(\mathbf{x})$ is linearly separable. Therefore, to evaluate the linear transformation described in Theorem 3.1, we assess

the degree to which the learned features can be classified linearly for data pairs (\mathbf{c}, \mathbf{x}) . Our initial experiments investigate the relationship between the degree of invertibility of the mapping from \mathbf{c} to \mathbf{x} and the approximation of the identifiability result in Theorem 3.1. This exploration provides empirical insights into how invertibility influences the recovery of latent variables. Specifically, we fix the size of the latent variables and gradually increase the size of the observed variables, thereby enhancing the degree of invertibility of the mapping from \mathbf{c} to \mathbf{x} . Detailed experimental setups are provided in Appendix G.1. The left of Figure 2 demonstrates that classification accuracy improves as the size of the observed variables \mathbf{x} increases, aligning with our theoretical results in Theorem 3.1.

We then examine the impact of latent graph structures on our identifiability results. To this end, we randomly generate DAG structures in the latent space. Specifically, random Erdős-Rényi (ER) graphs [24] are generated with varying numbers of expected edges. For instance, ER k denotes graphs with d nodes and kd expected edges. The right panel of Figure 2 illustrates the relationship between classification accuracy and the size of the latent variables \mathbf{c} under different settings, including ER1, ER2, and ER3. The results demonstrate that our identifiability findings hold consistently across various graph structures and latent variable sizes, as evidenced by the linear classification accuracy.

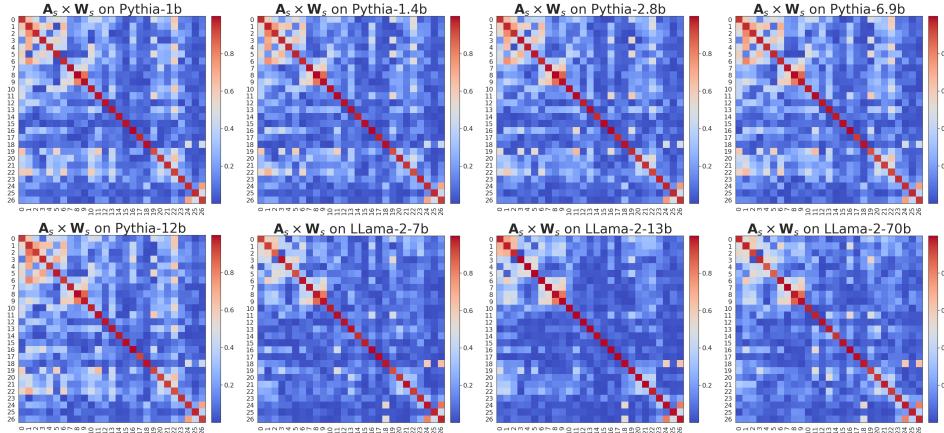


Figure 3: Results of the product $\mathbf{A}_s \times \mathbf{W}_s$ across the LLaMA-2 and Pythia model families. Here, \mathbf{A}_s represents a matrix derived from the feature differences of 27 counterfactual pairs, while \mathbf{W}_s is a weight matrix obtained from a linear classifier trained on these features. The product approximates the identity matrix, supporting the theoretical findings outlined in Corollary 4.3.

Experiments with LLMs We now present experiments on pre-trained LLMs. While it is challenging to collect all latent variables from real-world data to directly evaluate our identifiability result in Theorem 3.1, we can instead assess our corollaries to indirectly validate it. For Corollary 4.2, prior studies have already demonstrated the linear representation properties of LLM embeddings using counterfactual pairs that differ only in a single concept of interest [54, 60, 72, 43, 76, 59]. Therefore, we shift our focus to a new property highlighted in Corollary 4.3, specifically the relationship $\tilde{\mathbf{W}}\tilde{\mathbf{A}}^i \approx s\mathbf{I}$, which has not been explored in prior work. To investigate it, we utilize 27 counterfactual pairs from [59] that differ only in a *binary* concept, which are constructed based on the Big Analogy Test dataset [28]. More details can be found in Section G.2 in Appendix. Specifically, supported by

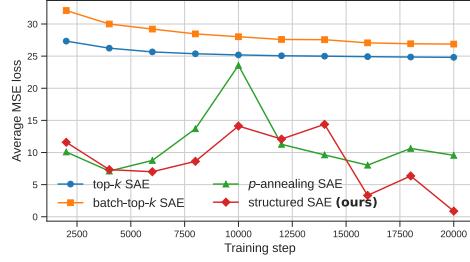
Figure 2: (Left) Classification accuracy under varying numbers of observed variables. (Right) Classification accuracy across different graph structures.

Corollary 4.2, we compute the differences between the representations of these 27 counterfactual pairs on pre-trained LLMs to construct a matrix \mathbf{A}_s with dimensions $27 \times \text{dim}$, where dim corresponds to the representation dimension of the pre-trained LLM used. \mathbf{A}_s denote a variant of $\tilde{\mathbf{A}}^i$, with each row is associated with the probability of a specific binary concept. See a binary special case of Corollary 4.2 in Appendix F for more details. Additionally, we train a linear classifier using the representations of these counterfactual pairs and extract the corresponding weights, supported by Corollary 4.3, forming a weight matrix \mathbf{W}_s with dimensions $\text{dim} \times 27$. Finally, after normalizing \mathbf{A}_s and \mathbf{W}_s to avoid scaling s in Corollary 4.3, we examine the product of \mathbf{A}_s and \mathbf{W}_s to assess whether $\mathbf{A}_s \mathbf{W}_s \approx \mathbf{I}$. Figure 3 displays the results of this product across the LLaMA-2 and Pythia model families. Refer to Appendix J for more results on LLaMA-3 and DeepSeek-R1. The results show that the product approximates the identity matrix, which is consistent with the theoretical finding in Corollary 4.3.

Experiments on Structured SAEs We train four sparse SAE variants—top- k SAE [26], batch-top- k SAE [17], p -annealing SAE [38], and the proposed structured SAE. Following Theorem 3.1, each SAE is trained on representations from the final hidden layer of the 70 million-parameter Pythia language model [8], with *The Pile* corpus [25]. For evaluation, we use 27 counterfactual pairs from [59] again to train a linear classification using LogisticRegression classifier from the scikit-learn library, to obtain logits, i.e., unnormalized $p(c^i = 1|\mathbf{x})$, for each of the 27 pairs. These counterfactual pairs are also passed through the trained SAEs to extract features \mathbf{z} . We search for the best-matching feature z_i for each $p(c^i|\mathbf{x})$ according to Pearson correlation between $\exp(z_i)$ and $p(c^i|\mathbf{x})$. Table 4a shows the Pearson correlations for a subset of 27 concepts, with additional results provided in Appendix H. These findings highlight the advantage of the proposed structured SAE, which benefits from incorporating structured regularization. This advantage is further evident in the reconstruction loss, as illustrated in Figure 4b.

Concepts	Pearson Correlation (\uparrow)			
	top- k	batch-top- k	p -annealing	Ours
3psg_ved	0.432	0.332	0.680	0.691
adj.adj.+ly	0.379	0.396	0.794	0.833
adj.comparative	0.425	0.334	0.592	0.727
average	0.365	0.331	0.638	0.674
country.capital	0.205	0.165	0.495	0.540
french_german	0.264	0.226	0.550	0.607
french_spanish	0.389	0.264	0.560	0.561
frequent_infrequent	0.262	0.255	0.472	0.554

(a) Pearson correlations across 4 SAEs.



(b) Reconstruction loss during training

Figure 4: Comparison of SAE models: correlation scores and reconstruction loss on the validation dataset.

7 Conclusion

Our analysis of large language models (LLMs) through the lens of latent variable models and identifiability offers key insights into the mechanisms driving their success³. We have demonstrated that the representations learned by LLMs can be effectively approximated as a linear transformation of the posterior distribution of latent variables. This not only provides an initial framework for understanding next-token prediction, but also highlights the underlying linear properties of LLMs, including support for the linear representation hypothesis and sparse autoencoders. These findings pave the way for further exploration of how LLMs learn and represent complex patterns in data. Based on this finding, we suggest the following directions: 1) Rethinking Invertibility Assumptions in Causal Representation Learning, 2) Embedding Causal Reasoning in LLMs Through Linear Unmixing. Refer to Appendix K for more details.

³We encourage the reader to refer to the additional discussions on latent variable models in Appendix I.

References

- [1] A. Acerbi and J. M. Stubbersfield. Large language models show human-like content biases in transmission chain experiments. *Proceedings of the National Academy of Sciences*, 120(44):e2313790120, 2023.
- [2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.
- [3] K. Ahuja, D. Mahajan, Y. Wang, and Y. Bengio. Interventional causal representation learning. In *ICML*, pages 372–407. PMLR, 2023.
- [4] S. Arora, R. Ge, T. Ma, and A. Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pages 113–149. PMLR, 2015.
- [5] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.
- [6] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.
- [7] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.
- [8] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.
- [9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [10] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [11] D. Braun, J. Taylor, N. Goldowsky-Dill, and L. Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. *Advances in Neural Information Processing Systems*, 37:107286–107325, 2024.
- [12] J. Brehmer, P. De Haan, P. Lippe, and T. Cohen. Weakly supervised causal representation learning. *arXiv preprint arXiv:2203.16437*, 2022.
- [13] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah. Towards monosematicity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [15] S. Buchholz, G. Rajendran, E. Rosenfeld, B. Aragam, B. Schölkopf, and P. Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. *arXiv preprint arXiv:2306.02235*, 2023.
- [16] C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

- [17] B. Bussmann, P. Leask, and N. Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.
- [18] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [19] B. Dumitrescu and P. Irofti. *Dictionary learning algorithms and applications*. Springer, 2018.
- [20] J. Eggert and E. Korner. Sparse coding and nmf. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 4, pages 2529–2533. IEEE, 2004.
- [21] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.
- [22] M. Elad and A. M. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.
- [23] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.
- [24] P. ERDdS and A. R&wi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.
- [25] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- [26] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.
- [27] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=tcsZt9ZNKD>.
- [28] A. Gladkova, A. Drozd, and S. 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.
- [29] Y. Gu and D. B. Dunson. Bayesian pyramids: Identifiable multilayer discrete latent structure models for discrete data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(2):399–426, 2023.
- [30] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [31] W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, and D. Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.
- [32] R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.
- [33] A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *NeurIPS*, 29, 2016.
- [34] A. Hyvarinen, H. Sasaki, and R. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.
- [35] Y. Jiang, G. Rajendran, P. Ravikumar, B. Aragam, and V. Veitch. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*, 2024.

- [36] S. Jin, S. Wiseman, K. Stratos, and K. Livescu. Discrete latent variable representations for low-resource text classification. *arXiv preprint arXiv:2006.06226*, 2020.
- [37] S. Kantamneni, J. Engels, S. Rajamanoharan, M. Tegmark, and N. Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.
- [38] A. Karvonen, B. Wright, C. Rager, R. Angell, J. Brinkmann, L. R. Smith, C. M. Verdun, D. Bau, and S. Marks. Measuring progress in dictionary learning for language model interpretability with board game models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=SCEdoGghcw>.
- [39] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *AISTAS*, pages 2207–2217. PMLR, 2020.
- [40] B. Kivva, G. Rajendran, P. Ravikumar, and B. Aragam. Learning latent causal graphs via mixture oracles. *Advances in Neural Information Processing Systems*, 34:18087–18101, 2021.
- [41] L. Kong, G. Chen, B. Huang, E. P. Xing, Y. Chi, and K. Zhang. Learning discrete concepts in latent hierarchical models. *arXiv preprint arXiv:2406.00519*, 2024.
- [42] S. C. Levinson. *Pragmatics*. Cambridge UP, 1983.
- [43] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.
- [44] Y. Liu, Z. Zhang, D. Gong, M. Gong, B. Huang, A. v. d. Hengel, K. Zhang, and J. Q. Shi. Identifying weight-variant latent causal models. *arXiv preprint arXiv:2208.14153*, 2022.
- [45] Y. Liu, Z. Zhang, D. Gong, M. Gong, B. Huang, A. v. d. Hengel, K. Zhang, and J. Q. Shi. Identifiable latent neural causal models. *arXiv preprint arXiv:2403.15711*, 2024.
- [46] Y. Liu, Z. Zhang, D. Gong, M. Gong, B. Huang, A. van den Hengel, K. Zhang, and J. Q. Shi. Identifiable latent polynomial causal models through the lens of change. In *The Twelfth International Conference on Learning Representations*, 2024.
- [47] Y. Liu, Z. Zhang, D. Gong, B. Huang, M. Gong, A. v. d. Hengel, K. Zhang, and J. Q. Shi. Revealing multimodal contrastive representation learning through latent partial causal models. *arXiv preprint arXiv:2402.06223*, 2024.
- [48] Y. Liu, Z. Zhang, D. Gong, M. Gong, B. Huang, A. van den Hengel, K. Zhang, and J. Q. Shi. Latent covariate shift: Unlocking partial identifiability for multi-source domain adaptation. *Transactions on Machine Learning Research*, 2025.
- [49] Q. Lyu, M. Apidianaki, and C. Callison-Burch. Representation of lexical stylistic features in language models’ embedding space. *arXiv preprint arXiv:2305.18657*, 2023.
- [50] C. D. Manning, K. Clark, J. Hewitt, U. Khandelwal, and O. Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020.
- [51] E. Marconato, S. Lachapelle, S. Weichwald, and L. Gresele. All or none: Identifiable linear properties of next-token predictors in language modeling. *arXiv preprint arXiv:2410.23501*, 2024.
- [52] S. Marks and M. Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.
- [53] R. Massidda, A. Geiger, T. Icard, and D. Bacciu. Causal abstraction with soft interventions. In *Conference on Causal Learning and Reasoning*, pages 68–87. PMLR, 2023.
- [54] T. Mikolov, W.-t. Yih, and G. 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, 2013.

- [55] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [56] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà. Relative representations enable zero-shot latent space communication. *arXiv preprint arXiv:2209.15430*, 2022.
- [57] N. Nanda, A. Lee, and M. Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.
- [58] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.
- [59] K. Park, Y. J. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.
- [60] J. Pennington, R. Socher, and C. 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.
- [61] S. Rajamanoharan, A. Conmy, L. Smith, T. Lieberum, V. Varma, J. Kramár, R. Shah, and N. Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024.
- [62] S. Rajamanoharan, T. Lieberum, N. Sonnerat, A. Conmy, V. Varma, J. Kramár, and N. Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.
- [63] G. Rajendran, S. Buchholz, B. Aragam, B. Schölkopf, and P. Ravikumar. Learning interpretable concepts: Unifying causal representation learning and foundation models. *arXiv preprint arXiv:2402.09236*, 2024.
- [64] G. Roeder, L. Metz, and D. Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pages 9030–9039. PMLR, 2021.
- [65] H. Sajjad, N. Durrani, F. Dalvi, F. Alam, A. R. Khan, and J. Xu. Analyzing encoded concepts in transformer language models. *arXiv preprint arXiv:2206.13289*, 2022.
- [66] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.
- [67] A. Seigal, C. Squires, and C. Uhler. Linear causal disentanglement via interventions. *arXiv preprint arXiv:2211.16467*, 2022.
- [68] X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang. Weakly supervised disentangled generative causal representation learning. *The Journal of Machine Learning Research*, 23(1):10994–11048, 2022.
- [69] C. Tigges, O. J. Hollinsworth, A. Geiger, and N. Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.
- [70] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [71] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [72] A. M. Turner, L. Thiergart, G. Leech, D. Udell, J. J. Vazquez, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308, 2023.
- [73] B. Varici, E. Acarturk, K. Shanmugam, A. Kumar, and A. Tajer. Score-based causal representation learning with interventions. *arXiv preprint arXiv:2301.08230*, 2023.

- [74] J. Von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Schölkopf, M. Besserve, and F. Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *NeurIPS*, 2021.
- [75] J. von Kügelgen, M. Besserve, L. Wendong, L. Gresele, A. Kekić, E. Bareinboim, D. Blei, and B. Schölkopf. Nonparametric identifiability of causal representations from unknown interventions. *Advances in Neural Information Processing Systems*, 36, 2023.
- [76] Z. Wang, L. Gui, J. Negrea, and V. Veitch. Concept algebra for score-based conditional model. In *ICML 2023 Workshop on Structured Probabilistic Inference \& Generative Modeling*, 2023.
- [77] H. Yan, L. Kong, L. Gui, Y. Chi, E. Xing, Y. He, and K. Zhang. Counterfactual generation with identifiability guarantees. *Advances in Neural Information Processing Systems*, 36, 2024.
- [78] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.
- [79] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.
- [80] J. Zheng and M. Meister. The unbearable slowness of being: Why do we live at 10 bits/s? *Neuron*, 2024.

Appendix

Table of Contents

A	Related Work	16
B	Limitations	16
C	Proof of Theorem 3.1	18
D	Proof of Corollary 4.2	21
E	Proof of Corollary 4.3	23
F	Extension of Corollaries 4.2 and 4.3 for a Binary Concept	24
F.1	Extension of Corollary 4.2 for a Binary Concept	24
F.2	Extension of Corollary 4.3 for a Binary Concept	25
G	Experimental Details Supporting Theoretical Results	26
G.1	Simulation Details	26
G.2	Experimental Details on LLMs	26
H	Experiment on Sparse Autoencoders	28
H.1	Implementation of the Proposed Structured SAE	28
H.2	Details of Evaluation Metric	28
H.3	Experiments and Results	29
I	Further Discussion: Observations on LLMs and World Models	31
I.1	LLMs Mimic the Human World Model, Not the World Itself	31
I.2	LLMs Versus Pure Vision Models: A Fundamental Difference	32
J	More Results on Llama-3 and DeepSeek-R1	33
K	Future Directions	34

A Related Work

Linearity of Representation in LLMs Recent studies have established the empirical finding that concepts in LLMs are often linearly encoded, a phenomenon known as the linear representation hypothesis [54, 60, 5, 23, 16, 69, 57, 56, 59, 43, 31, 63]. Building on this observation, recent works [59, 51] attempt to unify these findings into a cohesive framework, aiming to deepen our understanding and potentially inspire new insights. However, these works do not address why and how such linear properties emerge. Some previous works [51, 64] have demonstrated identifiability results within the inference space, such as establishing connections between features derived from different inference models. However, these results are confined to the inference space and do not connect to the true latent variables in latent variable models. A recent work [35] seeks to explain the origins of these linear properties but employs a latent variable model that differs from ours. From a technical perspective, their explanation is rooted in the implicit bias of gradient descent. In contrast, our work provides an explanation grounded in identifiability theory. This distinction highlights our focus on connecting the observed linear properties directly to the identifiability of true latent variables in latent variable models, offering a more comprehensive and theoretically robust understanding of these phenomena. In addition, recent work [63] assumes continuous latent and observed variables, while we assume both latent variables and observed variables to be discrete, aligning more closely with modeling natural language.

Causal Representation Learning This work is closely related to causal representation learning [66], which aims to identify high-level latent causal variables from low-level observational data. Many prior studies [12, 74, 53, 75, 3, 67, 68, 44, 15, 73, 46, 48, 45, 47, 33, 34, 39] have developed theoretical frameworks supporting the recovery of true latent variables up to simple transformations. However, these works primarily focus on continuous spaces and do not address the next-token prediction framework employed by LLMs. A subset of works [29, 41, 40] has explored causal representation learning in discrete spaces for both latent and observed variables, often imposing specific graph structures and assuming invertible mappings from latent to observed spaces. Despite their focus on discrete settings, none of these studies consider the next-token prediction framework in LLMs. In contrast, our work overcomes these limitations. We analyze approximate identifiability without relying on strict invertibility assumptions, which better aligns with the complex and often non-invertible relationships observed in real-world data. While a recent study has examined non-invertible mappings from latent to observed spaces, they rely on additional historical information to effectively restore invertibility. In our approach, we focus on achieving approximate identifiability without requiring such additional information. A very recent work [63] explores identifiability analysis for LLMs; however, they model both observed text data and latent variables as continuous and still assume invertibility in their analysis. In contrast, our framework explicitly considers discrete latent and observed variables, relaxed invertibility assumptions, and directly aligns with the next-token prediction paradigm, offering a more realistic and generalizable approach to causal representation learning in LLMs.

Sparse Autoencoders Polysemanticity, a phenomenon observed in recent studies, roughly speaking, refers to cases where a single representation encodes multiple distinct, human-interpretable concepts [6]. Early investigations suggested that neural networks represent features by linear superposition and motivated efforts to disentangle human-interpretable concepts from such linear mixing [23]. This can be achieved by using sparse autoencoders [32, 61, 62, 27, 11, 13], a technique closely related to the well-known framework of dictionary learning [19, 20, 21, 22, 2, 4]. In contrast to these works, we propose structured SAEs to model the dependencies among latent concepts. Furthermore, motivated by our theoretical findings, we introduce a new evaluation method for SAEs, grounded in our justified theoretical results.

B Limitations

One of the main contributions of this work is establishing an identifiability result for the next-token prediction framework, a widely used approach for training LLMs. This result hinges on a key assumption: the diversity condition, which requires the data distribution to exhibit sufficiently strong and diverse variations. The diversity condition was originally introduced in the context of nonlinear ICA [33, 34, 39] and has since been adopted in the identifiability analysis within the causal

representation learning community. In essence, it has emerged as a fundamental requirement for identifiability. This condition is likely satisfied given that the training data for LLMs is primarily sourced from the Internet, encompassing a broad and diverse range of content. Recent empirical analyses of LLMs further support the plausibility of this assumption [64, 51], reinforcing its relevance in the context of identifiability.

Similar to existing works [59, 35, 51, 63], our analysis is limited to a specific layer in LLMs and does not provide a justification for intermediate layers. Formalizing the relationship between features in intermediate layers and identifiability analysis presents additional challenges. This complexity arises from the intricate nature of intermediate layer representations. Therefore, extending identifiability results to intermediate layers remains an open and more challenging problem.

C Proof of Theorem 3.1

Proof. Next-token prediction mirrors multinomial logistic regression, where the model approximates the true conditional distribution, i.e., $p(y|\mathbf{x})$, by minimizing the cross-entropy loss. In this framework, given sufficient training data, a sufficiently expressive architecture, and effective optimization, the model's predictions are expected to converge to the true conditional distribution, as follows:

$$p(y|\mathbf{x}) = \frac{\exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y))}{\sum_{y'} \exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y'))}. \quad (7)$$

On the other hand, for the proposed latent variable model described in Eq. 1), the true conditional distribution $p(y|\mathbf{x})$ can be derived using Bayes' rule:

$$p(y|\mathbf{x}) = \sum_{\mathbf{c}} p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}). \quad (8)$$

By comparing Eq. (8) and Eq. (7), we arrive at the following relationship:

$$\frac{\exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y))}{\sum_{y'} \exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y'))} = \sum_{\mathbf{c}} p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}). \quad (9)$$

Taking the logarithm on both sides of Eq. (9), we obtain:

$$\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y) - \log Z(\mathbf{x}) = \log \left(\sum_c p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}) \right) \quad (10)$$

where $Z(\mathbf{x}) = \sum_{y'} \exp(\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y'))$ represents the normalization constant. This transformation provides a direct link between the representations $\mathbf{f}(\mathbf{x})$ learned by next-token prediction (Eq. (7)) and the true posterior distribution $p(\mathbf{c}|\mathbf{x})$.

Now, let us focus on the right of Eq. (10), we can obtain that:

$$\log \left(\sum_c p(y|\mathbf{c})p(\mathbf{c}|\mathbf{x}) \right) = \log p(y|\mathbf{x}), \quad (11)$$

$$= \mathbb{E}_{p(\mathbf{c}|y)} [\log p(y|\mathbf{x})], \quad (12)$$

$$= \mathbb{E}_{p(\mathbf{c}|y)} [\log \left(p(\mathbf{c}|y, \mathbf{x}) \frac{p(y|\mathbf{x})}{p(\mathbf{c}|y, \mathbf{x})} \right)], \quad (13)$$

$$= \mathbb{E}_{p(\mathbf{c}|y)} [\log p(y, \mathbf{c}|\mathbf{x})] - \mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c}|y, \mathbf{x})], \quad (14)$$

$$= \mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c}|\mathbf{x})] + \mathbb{E}_{p(\mathbf{c}|y)} [\underbrace{\log p(y|\mathbf{c}, \mathbf{x})}_{p(y|\mathbf{c})}] - \mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c}|y, \mathbf{x})], \quad (15)$$

where $p(y|\mathbf{c}, \mathbf{x}) = p(y|\mathbf{c})$, due to the conditional independence of y and \mathbf{x} given \mathbf{c} , as defined in the proposed latent variable model.

Together with Eq. (15) and Eq. (10), we have:

$$\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y) - \log Z(\mathbf{x}) = \mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c}|\mathbf{x})] + \underbrace{\mathbb{E}_{p(\mathbf{c}|y)} [\log p(y|\mathbf{c})]}_{b_y} - \mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c}|y, \mathbf{x})]. \quad (16)$$

Define $\mathbb{E}_{p(\mathbf{c}|y)} [\log p(y|\mathbf{c})] = b_y$, and define a vector $[p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i}$ as the vector constructed from the probabilities of all possible values of \mathbf{c} , conditional on \mathbf{x} , where $\mathbf{c}_i \in \mathcal{C}$ and \mathcal{C} is the set of all possible values that \mathbf{c} can take. As a result, $\mathbb{E}_{p(\mathbf{c}|y)} \log p(\mathbf{c}|\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{c}|y) \log p(\mathbf{c}|\mathbf{x}) = [p(\mathbf{c} = \mathbf{c}_i|y)]_{\mathbf{c}_i} [\log p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i}$, and similarly $\mathbb{E}_{p(\mathbf{c}|y)} \log p(\mathbf{c}|y, \mathbf{x}) = [p(\mathbf{c} = \mathbf{c}_i|y)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i|y, \mathbf{x})]_{\mathbf{c}_i}$. Then we can re-write Eq. (16) as follows:

$$\mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y) - \log Z(\mathbf{x}) = [p(\mathbf{c} = \mathbf{c}_i|y)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i|\mathbf{x})]_{\mathbf{c}_i} \quad (17)$$

$$- [p(\mathbf{c} = \mathbf{c}_i|y)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i|y, \mathbf{x})]_{\mathbf{c}_i} + b_y \quad (18)$$

Now, we apply the diversity condition outlined in Section 3, which asserts that there exist $\ell + 1$ values of y , i.e., y_0, y_1, \dots, y_ℓ , such that the matrix $\mathbf{L} = ([p(\mathbf{c} = \mathbf{c}_i|y = y_1)]_{\mathbf{c}_i}^T - [p(\mathbf{c} = \mathbf{c}_i|y = y_0)]_{\mathbf{c}_i}^T, \dots, [p(\mathbf{c} = \mathbf{c}_i|y = y_\ell)]_{\mathbf{c}_i}^T - [p(\mathbf{c} = \mathbf{c}_i|y = y_0)]_{\mathbf{c}_i}^T)$

$y_0)]_{\mathbf{c}_i}^T, \dots, [p(\mathbf{c} = \mathbf{c}_i | y = y_\ell)]_{\mathbf{c}_i}^T - [p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}^T \) of size $\ell \times \ell$ is invertible, where ℓ is the number of all possible values of \mathbf{c} . In this context, for $y = 0$, we have:$

$$\begin{aligned} \mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y = y_0) - \log Z(\mathbf{x}) &= [p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} \\ &\quad - \underbrace{[p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i | y = y_0, \mathbf{x})]_{\mathbf{c}_i}}_{h_{y_0}} + b_{y_0}, \end{aligned} \quad (19)$$

where we define $h_{y_0} = [p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i | y = y_0, \mathbf{x})]_{\mathbf{c}_i}$. For $y = 1$, we similarly obtain:

$$\begin{aligned} \mathbf{f}_x(\mathbf{x})^T \mathbf{f}_y(y = y_1) - \log Z(\mathbf{x}) &= [p(\mathbf{c} = \mathbf{c}_i | y = y_1)]_{\mathbf{c}_i}^T [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} \\ &\quad - h_{y_1} + b_{y_1}. \end{aligned} \quad (20)$$

Subtracting Eq. (19) from Eq. (20), we get the following expression:

$$\begin{aligned} (\mathbf{f}_y(y = y_1)^T - \mathbf{f}_y(y = y_0)^T) \mathbf{f}_x(\mathbf{x}) &= ([p(\mathbf{c} = \mathbf{c}_i | y = y_1) - p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}^T) [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} \\ &\quad - (h_{y_1} - h_{y_0}) + b_{y_1} - b_{y_0}. \end{aligned} \quad (21)$$

According to the diversity condition, where y can take $\ell+1$ values, we can obtain a total of ℓ equations similar to Eq. (21). Collecting all of these equations, we have:

$$\underbrace{(\mathbf{f}_y(y = y_1) - \mathbf{f}_y(y = y_0), \dots, \mathbf{f}_y(y = y_\ell) - \mathbf{f}_y(y = y_0))^T}_{\hat{\mathbf{L}}^T} \mathbf{f}_x(\mathbf{x}) \quad (22)$$

$$\begin{aligned} &= \underbrace{([p(\mathbf{c} = \mathbf{c}_i | y = y_1) - p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i}, \dots, [p(\mathbf{c} = \mathbf{c}_i | y = y_\ell) - p(\mathbf{c} = \mathbf{c}_i | y = y_0)]_{\mathbf{c}_i})^T}_{\mathbf{L}} \times \\ &\quad [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} - \underbrace{[h_{y_1} - h_{y_0}, \dots, h_{y_\ell} - h_{y_0}]}_{\mathbf{h}_y} + \underbrace{[b_{y_1} - b_{y_0}, \dots, b_{y_\ell} - b_{y_0}]}_{\mathbf{b}_y} \end{aligned} \quad (23)$$

According to the diversity condition, the matrix $\hat{\mathbf{L}}$ of size $\ell \times \ell$ is invertible, as a result, we arrive:

$$\mathbf{f}_x(\mathbf{x}) = \underbrace{(\hat{\mathbf{L}}^T)^{-1} \mathbf{L}}_{\mathbf{A}} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} - (\hat{\mathbf{L}}^T)^{-1} \mathbf{h}_y + (\hat{\mathbf{L}}^T)^{-1} \mathbf{b}_y. \quad (24)$$

Here due to the diversity condition, the matrix $\mathbf{L} = (p(\mathbf{c} | y = y_1) - p(\mathbf{c} | y = y_0), \dots, p(\mathbf{c} | y = y_\ell) - p(\mathbf{c} | y = y_0))$ of size $\ell \times \ell$ is also invertible, \mathbf{A} is invertible.

We first focus on the term \mathbf{b}_y on the right-hand side of Eq. (24). Note that $b_{y_i} = \mathbb{E}_{p(\mathbf{c} | y_i)}[\cdot]$ is a constant with respect to \mathbf{c} , as the expectation integrates over all possible values of \mathbf{c} . As a result, the entire term $(\hat{\mathbf{L}}^T)^{-1} \mathbf{b}_y$, denoted as \mathbf{b} , on the right-hand side of Eq. (24) is also a constant.

We then focus on the term \mathbf{h}_y in the right of Eq. (24), i.e., \mathbf{h}_y , from the viewpoints of when the mapping \mathbf{g} from latent space, i.e., \mathbf{c} , to observed space, i.e., \mathbf{x} and y , is invertible and approximately invertible, respectively.

Invertible When the mapping \mathbf{g} from latent space to observed space is invertible, meaning that for

$$1 - p(\mathbf{c} = \mathbf{c}^* | \mathbf{x}, y) = \epsilon, \quad (25)$$

we have $\epsilon = 0$. Then, for \mathbf{h}_y , we analyze each component, i.e., $h_{y_i} - h_{y_0}$, where

$$h_{y_i} = \mathbb{E}_{p(\mathbf{c} | y = y_i)} [\log p(\mathbf{c} | \mathbf{x}, y = y_i)], \quad h_{y_0} = \mathbb{E}_{p(\mathbf{c} | y = y_0)} [\log p(\mathbf{c} | \mathbf{x}, y = y_0)]. \quad (26)$$

When $\epsilon = 0$, the posterior distribution $p(\mathbf{c} | \mathbf{x}, y)$ becomes a delta distribution centered at \mathbf{c}^* , i.e.,

$$p(\mathbf{c} | \mathbf{x}, y) = \delta(\mathbf{c} - \mathbf{c}^*), \quad (27)$$

which implies that the posterior is concentrated at a single point \mathbf{c}^* , satisfying $\log p(\mathbf{c}^* | \mathbf{x}, y) = 0$. In this case, we have

$$h_{y_i} - h_{y_0} = 0. \quad (28)$$

Approximately Invertible When the mapping \mathbf{g} is approximately invertible, i.e., $\epsilon \rightarrow 0$, for \mathbf{h}_y , we analyze the difference:

$$h_{y_i} - h_{y_0} = \mathbb{E}_{p(\mathbf{c}|y=y_i)} [\log p(\mathbf{c} | \mathbf{x}, y_i)] - \mathbb{E}_{p(\mathbf{c}|y=y_0)} [\log p(\mathbf{c} | \mathbf{x}, y_0)]. \quad (29)$$

Since the generative map $\mathbf{g} : \mathbf{c} \mapsto (\mathbf{x}, y)$ is approximately invertible, the posterior $p(\mathbf{c} | \mathbf{x}, y)$ becomes sharply concentrated at a unique $\mathbf{c}^*(\mathbf{x}, y)$. That is, for any small $\epsilon > 0$, the posterior takes the form:

$$p(\mathbf{c} | \mathbf{x}, y) = (1 - \epsilon) \cdot \delta_{\mathbf{c}^*} + \epsilon \cdot q(\mathbf{c}), \quad (30)$$

where $\delta_{\mathbf{c}^*}$ is a point mass at $\mathbf{c}^*(\mathbf{x}, y)$, and $q(\mathbf{c})$ is a distribution over other latent codes with bounded support and $\max_{\mathbf{c} \neq \mathbf{c}^*} \log p(\mathbf{c} | \mathbf{x}, y) \leq \log \epsilon$.

Then for each expectation:

$$\mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c} | \mathbf{x}, y)] = p(\mathbf{c}^* | y) \log(1 - \epsilon) + \sum_{\mathbf{c} \neq \mathbf{c}^*} p(\mathbf{c} | y) \log p(\mathbf{c} | \mathbf{x}, y). \quad (31)$$

Now note:

- $\log(1 - \epsilon) \approx 0$,
- $\log p(\mathbf{c} | \mathbf{x}, y) \leq \log \epsilon$ for all $\mathbf{c} \neq \mathbf{c}^*$,
- and for $\mathbf{c} \neq \mathbf{c}^*$ $p(\mathbf{c} | y)$ is bounded.

So the entire expectation satisfies:

$$\mathbb{E}_{p(\mathbf{c}|y)} [\log p(\mathbf{c} | \mathbf{x}, y)] \approx \log \epsilon, \quad (32)$$

which do not depend on y . Therefore,

$$h_{y_i} - h_{y_0} \xrightarrow{\epsilon \rightarrow 0} 0. \quad (33)$$

□

D Proof of Corollary 4.2

Proof. We first prove that: when $\epsilon_{\mathbf{x}} \rightarrow 0$, i.e., when $p(\mathbf{c} | \mathbf{x})$ becomes sharply peaked at \mathbf{c}_x^* , we can approximate:

$$p(\mathbf{c}) \approx p(c^i | \mathbf{x}) \cdot p(\mathbf{c}^{-i} | \mathbf{x}), \quad (34)$$

where \mathbf{c}^{-i} denotes all concepts except c^i . To this end, we analysis their KL(Kullback–Leibler) residual term:

$$\text{Residual} := D_{\text{KL}}(p(\mathbf{c} | \mathbf{x}) \| p(c^i | \mathbf{x})p(\mathbf{c}^{-i} | \mathbf{x})). \quad (35)$$

Since when $\epsilon_{\mathbf{x}} \rightarrow 0$, $p(\mathbf{c} | \mathbf{x})$ is sharply peaked at a particular configuration $\mathbf{c}_x^* = (c^{i*}, \mathbf{c}^{-i*})$, i.e.:

$$p(\mathbf{c} | \mathbf{x}) = \begin{cases} 1 - \epsilon_{\mathbf{x}}, & \text{if } \mathbf{c} = \mathbf{c}_x^*, \\ \epsilon_{\mathbf{x}} \cdot r(\mathbf{c}), & \text{otherwise,} \end{cases} \quad (36)$$

where $\sum_{\mathbf{c} \neq \mathbf{c}_x^*} r(\mathbf{c}) = 1$.

Main Term in KL The KL divergence is defined as:

$$D_{\text{KL}}(p(\mathbf{c} | \mathbf{x}) \| p(c^i | \mathbf{x})p(\mathbf{c}^{-i} | \mathbf{x})) = \sum_{\mathbf{c}} p(\mathbf{c} | \mathbf{x}) \log \frac{p(\mathbf{c} | \mathbf{x})}{p(c^i | \mathbf{x})p(\mathbf{c}^{-i} | \mathbf{x})}. \quad (37)$$

The main contribution comes from $\mathbf{c} = \mathbf{c}_x^*$:

$$(1 - \epsilon_{\mathbf{x}}) \log \frac{1 - \epsilon_{\mathbf{x}}}{p(c^{i*} | \mathbf{x}) \cdot p(\mathbf{c}^{-i*} | \mathbf{x})}. \quad (38)$$

Note that:

$$p(c^{i*} | \mathbf{x}) = \sum_{\mathbf{c}^{-i}} p(c^{i*}, \mathbf{c}^{-i} | \mathbf{x}) \geq p(c^{i*}, \mathbf{c}^{-i*} | \mathbf{x}) = 1 - \epsilon_{\mathbf{x}}, \quad (39)$$

and similarly:

$$p(\mathbf{c}^{-i*} | \mathbf{x}) \geq 1 - \epsilon_{\mathbf{x}}. \quad (40)$$

Hence:

$$p(c^{i*} | \mathbf{x}) \cdot p(\mathbf{c}^{-i*} | \mathbf{x}) \geq (1 - \epsilon_{\mathbf{x}})^2, \quad (41)$$

$$\Rightarrow \frac{1 - \epsilon_{\mathbf{x}}}{p(c^{i*} | \mathbf{x}) \cdot p(\mathbf{c}^{-i*} | \mathbf{x})} \leq \frac{1}{1 - \epsilon_{\mathbf{x}}}. \quad (42)$$

Thus, the main term becomes:

$$(1 - \epsilon_{\mathbf{x}}) \log \left(\frac{1}{1 - \epsilon_{\mathbf{x}}} \right) = -(1 - \epsilon_{\mathbf{x}}) \log(1 - \epsilon_{\mathbf{x}}). \quad (43)$$

Tail Term For $\mathbf{c} \neq \mathbf{c}_x^*$, the contribution to the KL divergence is:

$$\epsilon_{\mathbf{x}} r(\mathbf{c}) \log \left(\frac{\epsilon_{\mathbf{x}} r(\mathbf{c})}{\epsilon_{\mathbf{x}}^2 r(c^i) r(\mathbf{c}^{-i})} \right) = \epsilon_{\mathbf{x}} r(\mathbf{c}) \log \left(\frac{1}{\epsilon_{\mathbf{x}}} \cdot \frac{r(\mathbf{c})}{r(c^i) r(\mathbf{c}^{-i})} \right). \quad (44)$$

Summing over all $\mathbf{c} \neq \mathbf{c}_x^*$, we obtain:

$$\epsilon_{\mathbf{x}} \sum_{\mathbf{c} \neq \mathbf{c}_x^*} r(\mathbf{c}) \log \left(\frac{1}{\epsilon_{\mathbf{x}}} \cdot \frac{r(\mathbf{c})}{r(c^i) r(\mathbf{c}^{-i})} \right) = \epsilon_{\mathbf{x}} \log \frac{1}{\epsilon_{\mathbf{x}}} + \epsilon_{\mathbf{x}} \sum_{\mathbf{c}} r(\mathbf{c}) \log \left(\frac{r(\mathbf{c})}{r(c^i) r(\mathbf{c}^{-i})} \right). \quad (45)$$

Since the second term is bounded, we conclude:

$$\text{Tail Term} = o(\epsilon_{\mathbf{x}} \log \epsilon_{\mathbf{x}}). \quad (46)$$

which vanishes faster than the main term as $\epsilon_{\mathbf{x}} \rightarrow 0$.

Final Bound Combining both contributions in Main Term and Tail Term, we get:

$$D_{\text{KL}}(p(\mathbf{c} | \mathbf{x}) \| p(c^i | \mathbf{x}) \cdot p(\mathbf{c}^{-i} | \mathbf{x})) \leq -(1 - \epsilon_{\mathbf{x}}) \log(1 - \epsilon_{\mathbf{x}}) + o(\epsilon_{\mathbf{x}} \log \epsilon_{\mathbf{x}}). \quad (47)$$

Here when $\epsilon_{\mathbf{x}} \rightarrow 0$, $D_{\text{KL}}(p(\mathbf{c} | \mathbf{x}) \| p(c^i | \mathbf{x}) \cdot p(\mathbf{c}^{-i} | \mathbf{x})) \rightarrow 0$.

Based on Eq. (34), $[\log p(\mathbf{c} | \mathbf{x})]_{\mathbf{c}_i}$ can be rewritten as:

$$[\log p(\mathbf{c} | \mathbf{x})]_{\mathbf{c}_i} \approx \mathbf{B}^i [\log p(c^i | \mathbf{x})]_{c^i} + \mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x})]_{\mathbf{c}^{-i}}. \quad (48)$$

Here, \mathbf{B}^i and \mathbf{B}^{-i} are binary broadcasting matrices that expand the marginal log-probability vectors $[\log p(c^i | \mathbf{x})]_{c^i}$ and $[\log p(\mathbf{c}^{-i} | \mathbf{x})]_{\mathbf{c}^{-i}}$ to the full configuration space $[\log p(\mathbf{c} | \mathbf{x})]_{\mathbf{c}_i}$, respectively.

As a result, for pair \mathbf{x}_0 and \mathbf{x}_1 that differ only in the i -th concept variable c^i , their difference in representation space is:

$$\begin{aligned} & [\log p(\mathbf{c} | \mathbf{x}_1)]_{\mathbf{c}_i} - [\log p(\mathbf{c} | \mathbf{x}_0)]_{\mathbf{c}_i} \\ & \approx \mathbf{B}^i [\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c^i} + \mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x}_1) - \log p(\mathbf{c}^{-i} | \mathbf{x}_0)]_{\mathbf{c}^{-i}}. \end{aligned} \quad (49)$$

We now show that:

$$p(\mathbf{c}^{-i} | \mathbf{x}_1) \approx p(\mathbf{c}^{-i} | \mathbf{x}_0), \quad (50)$$

so the term $\mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x}_1) - \log p(\mathbf{c}^{-i} | \mathbf{x}_0)]_{\mathbf{c}^{-i}}$ in Eq. (49) vanishes.

Recall Eq. (36), we have

$$p(\mathbf{c} | \mathbf{x}) = \begin{cases} 1 - \epsilon_{\mathbf{x}}, & \text{if } \mathbf{c} = \mathbf{c}_x^* = (c^{i*}, \mathbf{c}^{-i*}), \\ \epsilon_{\mathbf{x}} \cdot r(\mathbf{c}), & \text{otherwise,} \end{cases} \quad (51)$$

Then, for \mathbf{c}^{-i} , we have:

$$p(\mathbf{c}^{-i} | \mathbf{x}) = \sum_{c^i} p(c^i, \mathbf{c}^{-i} | \mathbf{x}) = \begin{cases} 1 - \epsilon_{\mathbf{x}}, & \text{if } (c^i, \mathbf{c}^{-i}) = \mathbf{c}_x^*, \\ \sum_{c^i} \epsilon_{\mathbf{x}} \cdot r(c^i, \mathbf{c}^{-i}), & \text{otherwise.} \end{cases} \quad (52)$$

This yields:

$$p(\mathbf{c}^{-i*} | \mathbf{x}) = 1 - \epsilon_{\mathbf{x}} + \epsilon_{\mathbf{x}} \sum_{c^i \neq c^{i*}} r(c^i, \mathbf{c}^{-i*}) = 1 - \epsilon_{\mathbf{x}} + o(\epsilon_{\mathbf{x}}), \quad (53)$$

$$p(\mathbf{c}^{-i} | \mathbf{x}) = \epsilon_{\mathbf{x}} \sum_{c^i} r(c^i, \mathbf{c}^{-i}) = o(\epsilon_{\mathbf{x}}), \quad \text{for } \mathbf{c}^{-i} \neq \mathbf{c}^{-i*}. \quad (54)$$

whether $\mathbf{c}^{-i} = \mathbf{c}^{-i*}$ (the dominant configuration) or $\mathbf{c}^{-i} \neq \mathbf{c}^{-i*}$ (a non-dominant configuration), we have for all \mathbf{c}^{-i} ,

$$p(\mathbf{c}^{-i} | \mathbf{x}_1) = p(\mathbf{c}^{-i} | \mathbf{x}_0) + o(\epsilon_{\mathbf{x}}), \quad (55)$$

which implies:

$$\log p(\mathbf{c}^{-i} | \mathbf{x}_1) - \log p(\mathbf{c}^{-i} | \mathbf{x}_0) = \log \left(1 + \frac{o(\epsilon_{\mathbf{x}})}{p(\mathbf{c}^{-i} | \mathbf{x}_0)} \right) = o(\epsilon_{\mathbf{x}}). \quad (56)$$

Consequently,

$$\mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x}_1) - \log p(\mathbf{c}^{-i} | \mathbf{x}_0)]_{\mathbf{c}^{-i}} = o(\epsilon_{\mathbf{x}}) \rightarrow 0 \quad \text{as } \epsilon_{\mathbf{x}} \rightarrow 0. \quad (57)$$

Together with Eq. (49), we get:

$$\mathbf{f}_x(\mathbf{x}_1) - \mathbf{f}_x(\mathbf{x}_0) \approx \mathbf{A} \mathbf{B}^i ([\log p(c^i | \mathbf{x}_1)]_{c^i} - [\log p(c^i | \mathbf{x}_0)]_{c^i}). \quad (58)$$

□

E Proof of Corollary 4.3

Again, when $\epsilon_x \rightarrow 0$, i.e., when $p(\mathbf{c} | \mathbf{x})$ becomes sharply peaked at \mathbf{c}_x^* , we can approximate:

$$p(\mathbf{c} | \mathbf{x}) \approx p(c^i | \mathbf{x}) \cdot p(\mathbf{c}^{-i} | \mathbf{x}), \quad (59)$$

Given the above, neglecting the constant term in the result in Theorem 3.1,

$$\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i}, \quad (60)$$

can be rewritten as

$$\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{\mathbf{c}_i} \approx \mathbf{A} (\mathbf{B}^i [\log p(c^i | \mathbf{x})]_{c^i} + \mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x})]_{\mathbf{c}^{-i}}). \quad (61)$$

In this case, for a data pair $(\mathbf{x}_0, \mathbf{x}_1)$ that differ only in the latent variable c^i , the representations $\mathbf{f}_x(\mathbf{x})$ are passed to a linear classifier with weights \mathbf{W} .

The classifier produces the logits:

$$\text{logits} \approx \mathbf{W} (\mathbf{A} (\mathbf{B}^i [\log p(c^i | \mathbf{x})]_{c^i} + \mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x})]_{\mathbf{c}^{-i}})). \quad (62)$$

For correct classification under cross entropy loss, the logits must match the true probabilities:

$$\text{logits} = s [\log p(c^i | \mathbf{x})]_{c^i}, \quad (63)$$

where s is a scaling, corresponding to normalization before applying softmax in cross entropy loss.

Thus, the weight matrix \mathbf{W} must satisfy the condition:

$$\mathbf{W} (\mathbf{A} \mathbf{B}^i) \approx s \mathbf{I}, \quad (64)$$

which ensures that the classifier produces the correct logits. Here \mathbf{I} denotes the identify matrix. The cross-entropy loss function is minimized when the predicted probabilities match the true probabilities, which is achieved when the logits are correctly aligned with $\log p(\mathbf{c} | \mathbf{x})$. Therefore, the classifier weights \mathbf{W} are related to \mathbf{A} by the invertible relation $\mathbf{W} \mathbf{A} \approx \mathbf{I}$, ensuring that the learned features can be mapped back to the latent distribution, and the classifier can make accurate predictions.

F Extension of Corollaries 4.2 and 4.3 for a Binary Concept

When considering pair that differ only in a binary concept of interest, i.e., c^i , both Corollaries 4.2 and 4.3 can be further refined, yielding the following results.

F.1 Extension of Corollary 4.2 for a Binary Concept

Corollary F.1 (Binary Concept Direction). *Suppose that Theorem 3.1 holds, and let c^i be a binary concept variable, i.e., $c^i \in \{0, 1\}$. Let \mathbf{x}_0 and \mathbf{x}_1 be a pair of inputs that differ only in the i -th **binary** concept c^i , with $c^i = 0$ for \mathbf{x}_0 and $c^i = 1$ for \mathbf{x}_1 . Then, as $\epsilon_{\mathbf{x}} \rightarrow 0$ in Definition 4.1, the representation difference simplifies as:*

$$\mathbf{f}_x(\mathbf{x}_1) - \mathbf{f}_x(\mathbf{x}_0) \approx \tilde{\mathbf{A}}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c^i}) \approx \log p(c^i = 0 | \mathbf{x}_1) \cdot \tilde{\mathbf{A}}^i \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad (65)$$

$$\text{or, } \approx \log p(c^i = 1 | \mathbf{x}_0) \cdot \tilde{\mathbf{A}}^i \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad (66)$$

where $\tilde{\mathbf{A}}^i = \mathbf{AB}^i$, \mathbf{B}^i is a binary lifting matrix that broadcasts each entry of $[\log p(c^i | \mathbf{x})]_{c^i}$ to the corresponding index in $[\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i}$. This shows that changes in a binary concept are encoded in a specific direction in the representation space defined by $\tilde{\mathbf{A}}^i$.

Proof. Recall Eq. (36), the joint concept distribution conditioned on input \mathbf{x} follows a peaked structure:

$$p(\mathbf{c} | \mathbf{x}) = \begin{cases} 1 - \epsilon_{\mathbf{x}}, & \text{if } \mathbf{c} = \mathbf{c}_{\mathbf{x}}^*, \\ \epsilon_{\mathbf{x}} \cdot r(\mathbf{c}), & \text{otherwise,} \end{cases} \quad (67)$$

where $\mathbf{c}_{\mathbf{x}}^* = (c_{\mathbf{x}}^{i*}, \mathbf{c}^{-i*})$ is the dominant concept configuration for \mathbf{x} , and $r(\mathbf{c})$ is a normalized residual distribution over all non-dominant \mathbf{c} .

Let \mathbf{x}_0 and \mathbf{x}_1 differ only in the i -th binary concept variable c^i , with:

$$\mathbf{c}_{\mathbf{x}_0}^* = (0, \mathbf{c}^{-i*}), \quad \mathbf{c}_{\mathbf{x}_1}^* = (1, \mathbf{c}^{-i*}). \quad (68)$$

Then the marginal probabilities for c^i are:

For \mathbf{x}_1 :

$$p(c^i = 1 | \mathbf{x}_1) = (1 - \epsilon_{\mathbf{x}_1}) + \epsilon_{\mathbf{x}_1} \cdot \sum_{\mathbf{c}^{-i} \neq \mathbf{c}^{-i*}} r(1, \mathbf{c}^{-i}) = 1 - \epsilon_{\mathbf{x}_1} \cdot \alpha_1, \quad (69)$$

where $\alpha_1 := 1 - \sum_{\mathbf{c}^{-i} \neq \mathbf{c}^{-i*}} r(1, \mathbf{c}^{-i})$.

$$p(c^i = 0 | \mathbf{x}_1) = 1 - p(c^i = 1 | \mathbf{x}_1) = \epsilon_{\mathbf{x}_1} \cdot \alpha_1. \quad (70)$$

For \mathbf{x}_0 :

$$p(c^i = 0 | \mathbf{x}_0) = (1 - \epsilon_{\mathbf{x}_0}) + \epsilon_{\mathbf{x}_0} \cdot \sum_{\mathbf{c}^{-i} \neq \mathbf{c}^{-i*}} r(0, \mathbf{c}^{-i}) = 1 - \epsilon_{\mathbf{x}_0} \cdot \alpha_0, \quad (71)$$

where $\alpha_0 := 1 - \sum_{\mathbf{c}^{-i} \neq \mathbf{c}^{-i*}} r(0, \mathbf{c}^{-i})$.

$$p(c^i = 1 | \mathbf{x}_0) = 1 - p(c^i = 0 | \mathbf{x}_0) = \epsilon_{\mathbf{x}_0} \cdot \alpha_0. \quad (72)$$

Taking logarithmic, we have:

$$\log p(c^i = 1 | \mathbf{x}_1) = \log(1 - \epsilon_{\mathbf{x}_1} \cdot \alpha_1) \approx 0, \quad \text{as } \epsilon_{\mathbf{x}} \rightarrow 0 \quad (73)$$

$$\log p(c^i = 0 | \mathbf{x}_1) = \log(\epsilon_{\mathbf{x}_1} \cdot \alpha_1). \quad (74)$$

$$\log p(c^i = 0 | \mathbf{x}_0) = \log(1 - \epsilon_{\mathbf{x}_0} \cdot \alpha_0) \approx 0, \quad \text{as } \epsilon_{\mathbf{x}} \rightarrow 0 \quad (75)$$

$$\log p(c^i = 1 | \mathbf{x}_0) = \log(\epsilon_{\mathbf{x}_0} \cdot \alpha_0) \approx \log(\epsilon_{\mathbf{x}_1} \cdot \alpha_1) = \log p(c^i = 0 | \mathbf{x}_1), \quad \text{as } \epsilon_{\mathbf{x}} \rightarrow 0. \quad (76)$$

Then the vector difference:

$$[\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c^i} = \begin{bmatrix} \log p(c^i = 0 | \mathbf{x}_1) - \log p(c^i = 0 | \mathbf{x}_0) \\ \log p(c^i = 1 | \mathbf{x}_1) - \log p(c^i = 1 | \mathbf{x}_0) \end{bmatrix} \quad (77)$$

$$\approx \begin{bmatrix} \log p(c^i = 0 | \mathbf{x}_1) \\ -\log p(c^i = 1 | \mathbf{x}_0) \end{bmatrix} \quad (78)$$

$$\approx \log p(c^i = 0 | \mathbf{x}_1) \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (79)$$

Finally, the representation difference becomes:

$$\mathbf{f}_x(\mathbf{x}_1) - \mathbf{f}_x(\mathbf{x}_0) \approx \tilde{\mathbf{A}}^i ([\log p(c^i | \mathbf{x}_1) - \log p(c^i | \mathbf{x}_0)]_{c^i}) \approx \log p(c^i = 0 | \mathbf{x}_1) \cdot \tilde{\mathbf{A}}^i \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (80)$$

Here, note that: $\log p(c^i = 0 | \mathbf{x}_1) \approx \log p(c^i = 1 | \mathbf{x}_0)$ as shown in Eq. 76. \square

F.2 Extension of Corollary 4.3 for a Binary Concept

Corollary F.2 (Binary Concept Classification). *Suppose that Theorem 3.1 holds, i.e., $\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} = \mathbf{c}_i | \mathbf{x})]_{c_i} + \mathbf{b}$. Let \mathbf{x}_0 and \mathbf{x}_1 be pair data that differ only in the i -th binary concept variable c^i , with labels c^i , where $c^i = 0$ for \mathbf{x}_0 and $c^i = 1$ for \mathbf{x}_1 . Then when $\epsilon_x \rightarrow 0$ in Def. 4.1, the corresponding representations $(\mathbf{f}(\mathbf{x}_0), \mathbf{f}(\mathbf{x}_1))$ are linearly separable with a weight vector \mathbf{w} satisfying $\mathbf{w} \tilde{\mathbf{A}}^i \begin{bmatrix} 1 \\ -1 \end{bmatrix} \approx s$, where s accounts for softmax scaling. The corresponding logit is the unnormalized $p(c^i = 1 | \mathbf{x})$.*

Proof. When $\epsilon_x \rightarrow 0$, the posterior $p(\mathbf{c} | \mathbf{x})$ becomes sharply peaked at a unique mode \mathbf{c}_x^* . This implies a near-independence of c^i and \mathbf{c}^{-i} given \mathbf{x} , allowing us to write:

$$p(\mathbf{c} | \mathbf{x}) \approx p(c^i | \mathbf{x}) \cdot p(\mathbf{c}^{-i} | \mathbf{x}). \quad (81)$$

Taking logs and substituting into Theorem 3.1 (neglecting the bias term \mathbf{b}), we obtain:

$$\mathbf{f}_x(\mathbf{x}) \approx \mathbf{A} [\log p(\mathbf{c} | \mathbf{x})]_{\mathbf{c}} \approx \mathbf{A} (\mathbf{B}^i [\log p(c^i | \mathbf{x})]_{c^i} + \mathbf{B}^{-i} [\log p(\mathbf{c}^{-i} | \mathbf{x})]_{\mathbf{c}^{-i}}), \quad (82)$$

where \mathbf{B}^i and \mathbf{B}^{-i} denote the lifting operators that map the marginal log-probabilities of c^i and \mathbf{c}^{-i} into the joint log-probability vector space.

Define $\tilde{\mathbf{A}}^i := \mathbf{A} \mathbf{B}^i$. Then:

$$\mathbf{f}_x(\mathbf{x}) \approx \tilde{\mathbf{A}}^i [\log p(c^i | \mathbf{x})]_{c^i} + (\text{independent of } c^i). \quad (83)$$

Consider a linear classifier with weight vector \mathbf{w} applied to $\mathbf{f}_x(\mathbf{x})$:

$$\text{logit}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{f}_x(\mathbf{x}) \approx \mathbf{w} \tilde{\mathbf{A}}^i \begin{bmatrix} p(c^i = 0 | \mathbf{x}) \\ 1 - p(c^i = 0 | \mathbf{x}) \end{bmatrix} + \text{const.} \quad (84)$$

To align this with the (unnormalized) output required for cross-entropy classification, we demand:

$$\text{logit}(\mathbf{x}) \approx s \cdot p(c^i = 0 | \mathbf{x}) = s \cdot (1 - p(c^i = 1 | \mathbf{x})), \quad (85)$$

which implies:

$$s_0 - s_1 \approx s, \quad (86)$$

where $[s_0, s_1] = \mathbf{w} \tilde{\mathbf{A}}^i$.

Ignoring the constant, we complete the proof. \square

G Experimental Details Supporting Theoretical Results

G.1 Simulation Details

For the left side of Figure 2, which investigates the relationship between the degree of invertibility in the mapping from \mathbf{c} to \mathbf{x} and the approximation of the identifiability result in Theorem 3.1, we aim to exclude other uncertain factors that might affect the result. To achieve this, we keep the number of latent variables constant (i.e., 3) and ensure that the graph structure follows a chain structure. Based on this structure, we model the conditional probabilities of each variable, given its parents, using Bernoulli distributions. The parameters of these distributions are uniformly sampled from the interval [0.2, 0.8], which are then used to generate the latent variables. Subsequently, we apply a one-hot encoding to these samples to obtain one-hot formal representations. These one-hot samples are then randomly permuted. For 3 latent variables, there are 2^3 possible permutations, each corresponding to 3 observed binary variables, resulting in a total of $2^3 \times 3$ different observed binary variables. We then randomly sample from these observed variables, varying the sample size. For example, as shown on the left in Figure 2, we can select different variables as observed variables. Clearly, as the number of observed variables increases, the mutual information between observed and latent variables also increases. As a result, the degree of invertibility from latent to observed variables increases.

For the right side of Figure 2, we explore the robustness of our identifiability result in Theorem 3.1 with respect to both the graph structure and the size of the latent variables. To this end, we randomly generate DAG structures in the latent space using Erdős-Renyi (ER) graphs [24], where ER k denotes graphs with d nodes and kd expected edges. For each ER k configuration, we also vary the size of the latent variables from 4 to 8, allowing us to examine how the size of latent variables influences the identifiability results. In terms of the observed variables, we adapt the experimental setup from the left side of Figure 2 to determine the appropriate observed variable size for different latent variable sizes. This ensures that the degree of invertibility from the latent space to the observed space remains sufficiently high, a crucial factor for the accuracy of our identifiability analysis.

Throughout the simulation, we use the following: In each experiment, we randomly mask one observed variable x_i , and use the remaining observed variables to predict it. Specifically, the remaining variables and the corresponding mask matrix are used as inputs to an embedding layer. This embedding layer transforms the input into a high-dimensional feature representation. The generated embeddings are then passed through a Multi-Layer Perceptron (MLP)-based architecture to extract meaningful features, e.g., $f_{\mathbf{x}}(\mathbf{x})$. The MLP model consists of three layers, each with 256 hidden units. After each layer, we apply Batch Normalization unit to stabilize training and a ReLU nonlinear activation function to introduce nonlinearity. The final output of the MLP is used to predict the masked variable through a linear classification layer. This allows us to assess how well the model can predict missing or masked values based on the remaining observed variables. We employ the Adam optimizer with a learning rate of $1e - 4$. To ensure robustness and account for potential variability in the results, we conduct each experimental setting with five different runs, each initialized with a different random seed. This procedure helps mitigate the effects of random initialization and provides a more reliable evaluation of the model’s performance.

For evaluation, we use the LogisticRegression classifier from the scikit-learn library, which operates on the features extracted from the output of the MLP-based architecture described above.

G.2 Experimental Details on LLMs

Unlike in simulation studies, where we have access to the complete set of latent variables, in real-world scenarios, their true values remain inherently unknown. This limitation arises from the very nature of latent variables—they are unobserved and must be inferred indirectly from the data. As a consequence, we cannot directly validate the linear identifiability results established in Theorem 3.1, since such validation would require explicit knowledge of these latent variables.

However, we can instead verify Corollary 4.3, which is a direct consequence of Theorem 3.1. By doing so, we provide indirect empirical evidence supporting the theoretical identifiability results. To achieve this, we need collect counterfactual pairs of data instances that differ in controlled and specific ways. These counterfactual pairs are essential for testing the implications of our theory in the context of real-world data.

Generating such counterfactual pairs, however, presents significant challenges. First, the inherent complexity and nuances of natural language make it difficult to create pairs that differ in precisely the intended contexts while leaving other aspects unchanged. Second, as highlighted in prior works [59, 35], constructing such counterfactual sentences is a highly non-trivial task, even for human annotators, due to the intricacies of semantics and the need for precise control over contextual variations.

To overcome these challenges, we adopt an approach based on existing resources. Specifically, we utilize the 27 counterfactual pairs introduced in [59], which provide a structured and well-curated set of counterfactual pairs. These concepts encompass a wide range of semantic and morphological transformations, as detailed in Table 1. By leveraging this established dataset, we ensure consistency with previous research while facilitating a robust and meaningful evaluation of Corollary 4.3.

We first use these 27 counterfactual pairs to construct a \mathbf{A}_s with size $27 \times dim$ by using the differences in the representations of these 27 counterfactual pairs, where dim corresponds to the feature dimension of the used LLM, such as 4096 for the Llama-27B model. To construct the corresponding matrix \mathbf{W}_s , we train a linear classifier using these 27 counterfactual pairs. Specifically, we use the representations of the counterfactual pairs as input and the corresponding values of the latent variables as output. As a result, the corresponding linear weights for the 27 counterfactual pairs are used to create \mathbf{W}_s . In our experiments, we employ the LogisticRegression classifier from the scikit-learn library.

Note that, as mentioned in Corollary 4.3, there is a scale s in $\mathbf{A}_s \mathbf{W}_s \approx s\mathbf{I}$. Therefore, we normalize both \mathbf{A}_s and \mathbf{W}_s , before computing $\mathbf{A}_s \times \mathbf{W}_s$. The scaling indeterminacy of latent variables is an inherent property of the latent space. However, in most cases, e.g., classification, the exact scaling factor is not particularly significant. For example, concepts are often represented as directions rather than magnitudes.

Table 1: Concept names, one example of the counterfactual pairs, and the number of used pairs, taken from [59].

#	Concept	Example	Word Pair Counts
1	verb \Rightarrow 3pSg	(accept, accepts)	50
2	verb \Rightarrow Ving	(add, adding)	50
3	verb \Rightarrow Ved	(accept, accepted)	50
4	Ving \Rightarrow 3pSg	(adding, adds)	50
5	Ving \Rightarrow Ved	(adding, added)	50
6	3pSg \Rightarrow Ved	(adds, added)	50
7	verb \Rightarrow V + able	(accept, acceptable)	50
8	verb \Rightarrow V + er	(begin, beginner)	50
9	verb \Rightarrow V + tion	(compile, compilation)	50
10	verb \Rightarrow V + ment	(agree, agreement)	50
11	adj \Rightarrow un + adj	(able, unable)	50
12	adj \Rightarrow adj + ly	(according, accordingly)	50
13	small \Rightarrow big	(brief, long)	25
14	thing \Rightarrow color	(ant, black)	50
15	thing \Rightarrow part	(bus, seats)	50
16	country \Rightarrow capital	(Austria, Vienna)	158
17	pronoun \Rightarrow possessive	(he, his)	4
18	male \Rightarrow female	(actor, actress)	52
19	lower \Rightarrow upper	(always, Always)	73
20	noun \Rightarrow plural	(album, albums)	100
21	adj \Rightarrow comparative	(bad, worse)	87
22	adj \Rightarrow superlative	(bad, worst)	87
23	frequent \Rightarrow infrequent	(bad, terrible)	86
24	English \Rightarrow French	(April, avril)	116
25	French \Rightarrow German	(ami, Freund)	128
26	French \Rightarrow Spanish	(annee, año)	180
27	German \Rightarrow Spanish	(Arbeit, trabajo)	228

H Experiment on Sparse Autoencoders

H.1 Implementation of the Proposed Structured SAE

The proposed structured SAE employs two regularization terms: a structured regularization to model the dependence among latent concepts, and a sparsity regularization based on the assumption that latent concepts may be sparsely activated. We implement it as follows:

$$\mathbf{S} = \text{ReLU}(\mathbf{w}_s(\mathbf{f}_x(\mathbf{x}) - \mathbf{b}_d) + \mathbf{b}_s), \quad (87)$$

$$\mathbf{R} = \text{ReLU}(\mathbf{w}_l(\mathbf{f}_x(\mathbf{x}) - \mathbf{b}_d) + \mathbf{b}_l), \quad (88)$$

$$\mathbf{z} = \mathbf{S} + \mathbf{R}, \quad (89)$$

$$\bar{\mathbf{f}}(\mathbf{x}) = \mathbf{w}_d\mathbf{z} + \mathbf{b}_d. \quad (90)$$

Here:

- \mathbf{S} denotes the sparse representations from the sparse encoder (with parameters $\mathbf{w}_s, \mathbf{b}_s$);
- \mathbf{R} denotes the structured representation from the structured encoder (with parameters $\mathbf{w}_l, \mathbf{b}_l$);
- \mathbf{z} is the combined representations used for reconstruction;
- $\bar{\mathbf{f}}(\mathbf{x})$ denote the reconstruction of $\mathbf{f}(\mathbf{x})$.

The loss function for training is:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{train}}} [\|\mathbf{f}(\mathbf{x}) - \bar{\mathbf{f}}(\mathbf{x})\|_2^2 + \lambda_t (\|\mathbf{S}\|_{p_t}^{p_t} + \gamma \|\mathbf{R}\|_{\text{nuc}})], \quad (91)$$

where:

- λ_t is the dynamically adjusted sparsity coefficient at step t , following p -annealing SAE [38];
- γ is a hyperparameter that balances the sparsity penalty and the low-rank regularization;
- $\|\mathbf{S}\|_{p_t}^{p_t} = \sum_i |s_i|^{p_t}$ is the adaptive ℓ_{p_t} norm promoting sparsity, following p -annealing SAE [38];
- $\|\mathbf{R}\|_{\text{nuc}}$ is the nuclear norm, used to encourage low-rank structure.

We estimate the nuclear norm using the top- k singular values:

$$\|\mathbf{R}\|_{\text{nuc}} \approx \sum_{i=1}^k \sigma_i, \quad (92)$$

where $\{\sigma_i\}_{i=1}^k$ are the largest k singular values, obtained via low-rank SVD (e.g., PyTorch's `svd_lowrank`). The value of k is a tunable parameter (e.g., $k = 64$) that trades off approximation accuracy and computational cost.

H.2 Details of Evaluation Metric

Obtaining $p(c^i = 1|\mathbf{x})$ from Supervised Linear Classification. For evaluation, we first use 27 counterfactual pairs from [59], also see Table 1, to train a linear classification using LogisticRegression classifier from the scikit-learn library, to obtain logits, i.e., unnormalized $p(c^i = 1|\mathbf{x})$, for each of the 27 pairs. As a result, for each concept in these 27 concepts, we can obtain the corresponding logit. Stacking these 27 logits yields the logit vector

$$\mathbf{u} = (u_1, u_2, \dots, u_{27}). \quad (93)$$

Extracting z_i from trained SAEs. We use the representations $\mathbf{f}_x(\mathbf{x})$ of the same 27 counterfactual pairs from [59] as input to a trained SAE, extracting the corresponding latent features \mathbf{z} . Let $\tilde{\mathbf{z}}$ denote the element-wise exponentiation of \mathbf{z} , i.e., $\tilde{\mathbf{z}} = \exp(\mathbf{z})$. This yields a feature matrix of size $27 \times D$, where D is the dimensionality of \mathbf{z} :

$$\tilde{\mathbf{z}} = (\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_{27})^T. \quad (94)$$

Correlation Matrix and Assignment. For the logit vector \mathbf{u} , and the feature matrix $\tilde{\mathbf{z}}$, we compute the Pearson correlation

$$\mathbf{R}_d = \text{corr}(\mathbf{u}, \tilde{\mathbf{z}}_{:,d}), \quad d = 1, \dots, D \quad (95)$$

where $\tilde{\mathbf{z}}_{:,d}$ denotes the d -th column of $\tilde{\mathbf{z}}$.

Note that the estimated features \mathbf{z} from the SAE are subject to permutation indeterminacy. To address this, we apply the Hungarian algorithm to solve the assignment problem on \mathbf{R}_d . This yields the optimal assignment for each concept, allowing us to compute the assigned Pearson correlation. We report the mean Pearson correlation across the 27 concepts, as well as the percentage of concepts whose Pearson correlation exceeds a threshold of 0.5.

H.3 Experiments and Results

We train four SAE variants—top- k SAE [26], batch-top- k SAE [17], p -annealing SAE [38], and our proposed structured SAE. Each SAE is trained on activations from the final hidden layer ($n = 512$) of the 70 million-parameter Pythia language model [8], which was pre-trained on *The Pile* corpus [25].

Experimental setup. To obtain training data for the SAEs, we run the pretrained Pythia-70M model over a duplicated copy of *The Pile* and cache the representations of its last hidden layer. All SAE variants use the same feature dimension ($D = 32,768$) and are trained for 20 000 optimization steps with a batch size of 10 000. We employ the Adam optimizer with an initial learning rate of 1×10^{-4} and linearly warm up the learning rate during the first 200 steps. For the top- k and batch-top- k SAEs, we set $k = 32$. For p -annealing SAEs, we apply a sparsity warm-up of 400 steps and an initial sparsity penalty coefficient $\lambda_s = 0.1$. The p -annealing-LoRa SAE uses the same p -annealing settings and additionally applies a low-rank scaling factor $\gamma = 0.1$.

Compute resources. All experiments are conducted on a server equipped with four NVIDIA A100 GPUs (40 GB each) running CUDA 12.2 and driver version 535.161.07. The host system features an AMD EPYC 7313 (16 cores) CPU and 503 GB of RAM. Caching the last-layer activations of The Pile dataset requires 51.5 GPU-hours. Training the top- k SAE takes 2 hours, while each of the other SAE variants requires 3 hours.

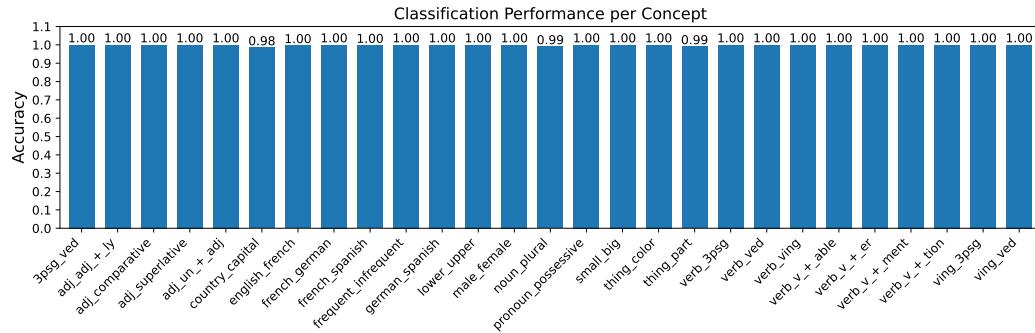


Figure 5: Classification accuracy of logistic probes across various concepts. Each bar represents the performance for a given concept.

Feature reconstruction. To quantify how well each sparse auto-encoder recovers the original hidden representations, we follow prior work and report the mean-squared error (MSE) between the SAE-reconstructed representations and the ground-truth representations for the 27 evaluation word-pairs. Figure 6 plots the validation MSE over training steps for all four variants. All methods reduce error monotonically, but the two annealing strategies clearly dominate the fixed- k baselines. Our p -annealing-LoRa SAE converges fastest and attains the lowest final MSE, followed by the plain p -annealing SAE. Both *top-k* and *Batch-top-k* remain larger than the annealing approaches throughout training.

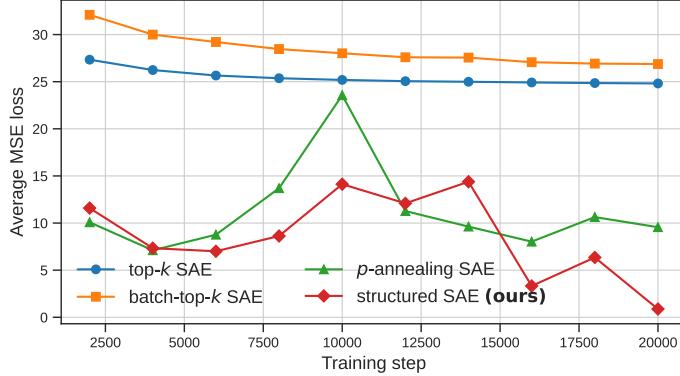


Figure 6: Validation reconstruction loss (**lower** is better) as training progresses.

Pearson Correlation Coefficient. We report Pearson Correlation Coefficient (PCC) at the 20 000-step checkpoint. As summarized in Table 2, the two p -annealing variants markedly outperform the fixed- k baselines. Across the 27 concepts, the proposed structured SAE achieves the highest mean PCC (0.674) and covers **96.4%** of concepts above the 0.5 threshold, followed closely by plain p -annealing (0.638/89.3%). In contrast, *top- k* and *batch-top- k* attain average PCCs of only 0.365 and 0.331, with fewer than 11% and 8% of concepts exceeding the 0.5 bar. The improvements are consistent: for 25 of the 27 concepts the strongest correlation is found in one of the annealing dictionaries, confirming that dynamic sparsity and the low-rank adaptation yield features that align more cleanly with human-interpretable concepts.

Ablation on the low-rank coefficient. We vary the low-rank term scaling factor $\gamma \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ while holding every other hyper-parameter fixed. The case $\gamma = 0$ reduces to the plain p -annealing SAE baseline, whereas larger γ values inject an increasingly strong low-rank update. At each checkpoint we record three diagnostics: (i) reconstruction error (MSE), (ii) mean PCC, i.e., PCC_{avg} , and (iii) The percentage of concepts whose Pearson correlation exceeds a threshold of 0.5, i.e., coverage@0.5. Figure 7 overlays the trajectories of these metrics across training steps. A moderate adapter strength ($\alpha \approx 10^{-1}$) consistently yields the lowest MSE and the highest interpretability scores, while too large or too small values degrade at least one objective, underscoring the importance of tuning α .

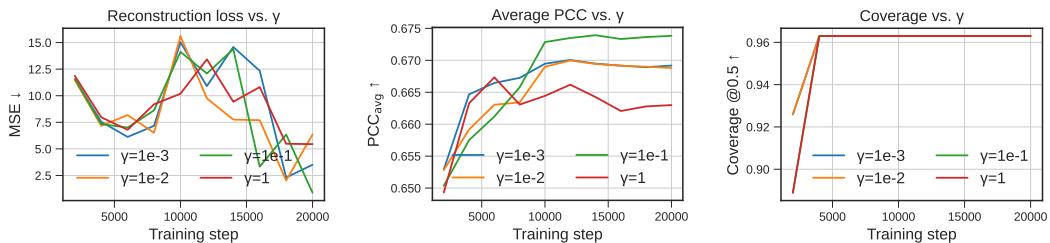


Figure 7: **Ablation study of the LoRa scaling coefficient γ .** Each curve plots a different $\gamma \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ over training: **(left)** reconstruction loss (MSE, lower is better); **(middle)** average Pearson Correlation Coefficient (PCC_{avg} , higher is better); **(right)** coverage@0.5 (higher is better).

Table 2: PCC results at training step 20,000. Each row reports the PCC for a single concept, followed by the overall average and coverage@0.5 across the 27 concepts. Bold values denote the best-performing method per row.

Concepts	Paired-concept correlation (PCC) across different SAEs (\uparrow)			
	top- k	batch-top- k	p -annealing	Our structured SAE
3psg_ved	0.432	0.332	0.680	0.691
adj_adj_+_ly	0.379	0.396	0.794	0.833
adj_comparative	0.425	0.334	0.592	0.727
adj_superlative	0.321	0.302	0.772	0.746
adj_un_+_adj	0.534	0.472	0.741	0.732
average	0.365	0.331	0.638	0.674
country_capital	0.205	0.165	0.495	0.540
english_french	0.291	0.265	0.666	0.649
french_german	0.264	0.226	0.550	0.607
french_spanish	0.389	0.264	0.560	0.561
frequent_infrequent	0.262	0.255	0.472	0.554
german_spanish	0.250	0.251	0.667	0.701
lower_upper	0.204	0.293	0.531	0.543
male_female	0.413	0.312	0.532	0.544
noun_plural	0.357	0.346	0.636	0.650
pronoun_possessive	0.725	0.779	0.998	0.991
small_big	0.327	0.225	0.506	0.547
thing_color	0.588	0.574	0.895	0.880
thing_part	0.240	0.207	0.400	0.453
verb_3psg	0.312	0.268	0.622	0.617
verb_v_+_able	0.458	0.361	0.763	0.806
verb_v_+_er	0.318	0.363	0.609	0.666
verb_v_+_ment	0.407	0.288	0.623	0.674
verb_v_+_tion	0.429	0.357	0.653	0.706
verb_ved	0.371	0.333	0.573	0.656
verb_ving	0.369	0.485	0.612	0.728
ving_3psg	0.286	0.233	0.646	0.711
ving_ved	0.304	0.245	0.627	0.682
Average	0.365	0.331	0.638	0.674
Coverage@0.5	10.7%	7.1%	89.3%	96.4%

I Further Discussion: Observations on LLMs and World Models

I.1 LLMs Mimic the Human World Model, Not the World Itself

As we explore the implications of our linear identifiability result, it is important to situate it within the broader context of human cognition—specifically, how humans develop and interact with an internal world model. In this subsection, we introduce the concept that LLMs Mimic the Human World Model, Not the World Itself, emphasizing the distinction between the vast physical world and the compressed abstraction humans use for reasoning and decision-making. Our analysis shows that LLMs replicate human-like abstractions through latent variable models. We further argue that LLMs aim to emulate this internal, compressed world model—rather than the physical world itself—by learning from human-generated text. This distinction provides critical insight into the success of LLMs in tasks aligned with human conceptualization and deepens our understanding of the relationship between language models and human cognition.

Humans gradually develop their understanding of the environment through learning from others and interacting with the world. This internal representation of our external environment is known as a world model. A key observation is that this model is not a direct reflection of the physical world but rather a highly compressed abstraction of it. For instance, numbers, such as 1, 2, and 3, are abstract tools created by the human mind for reasoning and problem-solving. They do not exist as tangible entities in the natural world. Similarly, many aspects of reality that escape human senses or even the most sophisticated scientific instruments are absent from our mental representations.

Interestingly, our texts reflect our mental activities and emotions, providing a window into this compressed world model. A recent study reveals a striking disparity between the limited information throughput of human behavior (approximately 10 bits/s) and the vast sensory input available (around 10^9 bits/s) [80]. This suggests that the human world model is an efficient, compressed abstraction, enabling us to reason, predict, and make decisions effectively. Despite this compression, humans have flourished as the dominant species on Earth, demonstrating the power of such a streamlined model.

LLMs aim to mimic not the vast and unbounded world but this human-compressed world model, which is significantly smaller and more manageable. By learning from human text, which encodes this abstraction, LLMs effectively replicate the patterns, reasoning, and abstractions that have proven successful for humans. This explains the impressive performance of LLMs in tasks that align with human understanding. Furthermore, the overlap between the latent space of LLMs (representing human concepts) and the observed space (human text) provides a powerful mechanism for aligning human-like abstractions with model predictions. For instance, modifying words in the observed space often corresponds to predictable changes in latent concepts.

I.2 LLMs Versus Pure Vision Models: A Fundamental Difference

While LLMs model human language, which has already undergone significant compression through human cognition, vision models face a fundamentally different challenge. In the previous subsection, we discussed how LLMs mimic the human world model, leveraging compressed abstractions derived from human-generated text. In contrast, vision models operate on raw, high-dimensional data from visual inputs. Moreover, the training data for vision models represents only a tiny fraction of the universe, constrained by the limitations of capturing devices and datasets. This vastness and lack of compression make the task of building generalizable representations in vision models inherently more complex than in NLP-based LLMs.

This difference in the nature of their training data and observed space might explain the behavioral differences between LLMs and pure vision models. While LLMs benefit from the inherent abstraction and compression of human language, vision models must contend with raw, unprocessed inputs that require far greater generalization capabilities. This underscores the importance of understanding the nuances of each modality when designing and evaluating AI systems.

J More Results on Llama-3 and DeepSeek-R1

We conduct additional experiments on recent LLMs, including Llama-3 and DeepSeek-R1, to further evaluate our findings. The experimental setup strictly adheres to the settings described in Section G.2. This enables a comprehensive investigation of our findings, ensuring that the results are thoroughly evaluated and validated across diverse LLM architectures and experimental conditions. Overall, we can see, the product $\mathbf{A}_s \times \mathbf{W}_s$ approximates the identity matrix, supporting the theoretical findings outlined in Corollary 4.3.

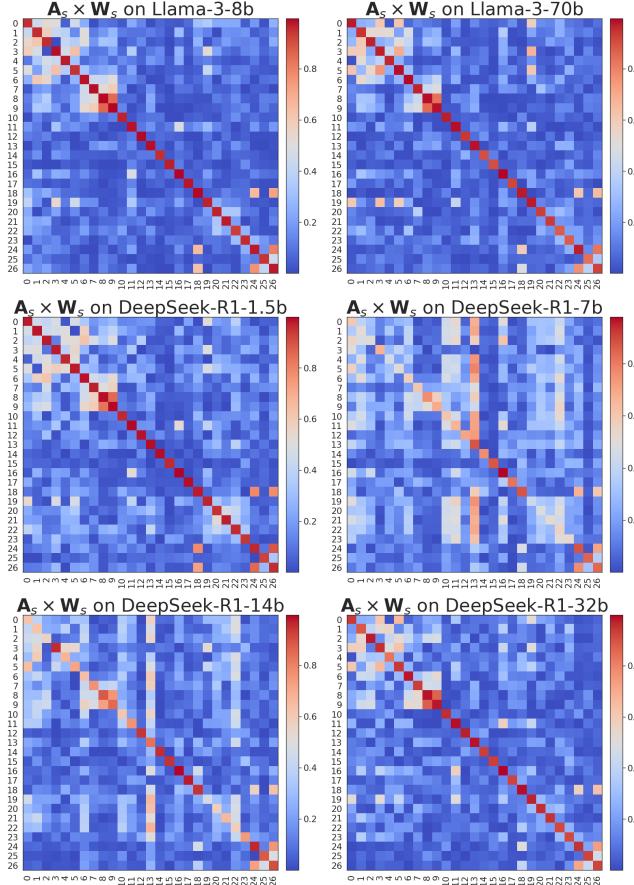


Figure 8: Results of the product $\mathbf{A}_s \times \mathbf{W}_s$ across the LLaMA-3 and DeepSeek-R1 model families. Here, \mathbf{A}_s represents a matrix derived from the feature differences of 27 counterfactual pairs, while \mathbf{W}_s is a weight matrix obtained from a linear classifier trained on these features.

K Future Directions

Rethinking Invertibility Assumptions in Causal Representation Learning Our identifiability analysis is closely related to the concept of identifiability in causal representation learning. Notably, to the best of our knowledge, this is the first work to explore approximate identifiability in the context of *non-invertible* mappings from latent space to observed space—a departure from the commonly upheld *invertibility* assumption in the causal representation learning community. We hope that our work will inspire future research aimed at overcoming the limitations imposed by invertibility assumptions in causal representation learning.

Embedding Causal Reasoning in LLMs Through Linear Unmixing Our linear identifiability result lays a foundation for uncovering latent causal relationships among concepts, especially when these variables exhibit causal dependencies within the proposed latent variable model. By showing that the representations in LLMs are linear mixtures of latent causal variables, our analysis shows that linear unmixing of these representations may allow for the identification of underlying latent causal structures. This approach not only opens up the possibility of understanding causal dynamics within LLMs but also suggests that causal reasoning, particularly in latent spaces, could be achievable through the exploration of these linear unmixing techniques. We believe this work marks a pivotal step toward embedding robust causal reasoning capabilities into LLMs.