LLM Interpretability with Identifiable Temporal-Instantaneous Representation

Xiangchen Song[†,1] **Jiaqi Sun**[†,1] **Zijian Li**[2] **Yujia Zheng**[1] **Kun Zhang**[1,2]
[1]Carnegie Mellon University
[2]Mohamed bin Zayed University of Artificial Intelligence
{xiangchensong,jiaqisun,kunz1}@cmu.edu

Abstract

Despite Large Language Models' remarkable capabilities, understanding their internal representations remains challenging. Mechanistic interpretability tools such as sparse autoencoders (SAEs) were developed to extract interpretable features from LLMs but lack *temporal dependency modeling*, *instantaneous relation representation*, and more importantly *theoretical guarantees*—undermining both the theoretical foundations and the practical confidence necessary for subsequent analyses. While causal representation learning (CRL) offers theoretically-grounded approaches for uncovering latent concepts, existing methods *cannot* scale to LLMs' rich conceptual space due to inefficient computation. To bridge the gap, we introduce an identifiable temporal causal representation learning framework specifically designed for LLMs' high-dimensional concept space, capturing both time-delayed and instantaneous causal relations. Our approach provides theoretical guarantees and demonstrates efficacy on synthetic datasets scaled to match real-world complexity. By extending SAE techniques with our temporal causal framework, we successfully discover meaningful concept relationships in LLM activations. Our findings show that modeling both temporal and instantaneous conceptual relationships advances the interpretability of LLMs.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language tasks, from question answering to content generation. Despite these achievements, a fundamental understanding of their internal representations remains underexplored. This gap between performance and interpretability poses significant challenges for ensuring the reliability, safety, and appropriate deployment of these increasingly powerful systems [48].

Mechanistic interpretability (MI) aims to bridge this gap by reverse-engineering neural networks to understand how they process and represent information [14]. Among all MI tools, sparse autoencoders (SAEs) have emerged as a promising approach for extracting interpretable features from LLMs [14, 54]. By decomposing the high-dimensional activations of LLMs into sparse, monosemantic features, SAEs help identify the basic units of computation within these complex systems. However, SAEs present several limitations that restrict their utility for comprehensive model understanding:

First, SAEs treat each feature as an isolated representation, failing to capture how features influence one another. This omission disregards semantic connections and transitions within a sequence, which are known as temporal or time-delayed relationships between features[*]. Second, SAEs lack mechanisms to represent instantaneous or logical relationships between features, such as mutual

[†]Equal contribution.
[*]Alternative approaches, such as [1, 31], use attention scores from the LLM to infer time-delayed influence.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

exclusivity or co-occurrence constraints [33]. These relationships complement the temporal dynamics by encoding structural dependencies within the same time step. Third, and most critically, SAEs offer no theoretical guarantees of the uniqueness of the recovered features. This absence undermines confidence that the extracted features reflect meaningful and stable latent variables, rather than arbitrary or unstable transformations [56].

Fortunately, to address these limitations, the causal representation learning (CRL) community has proposed a range of promising frameworks with theoretical guarantees [48]. For instance, [28] and [32] use sparse causal influence and interventions to uncover temporal and instantaneous relationships among latent variables. However, these methods face significant scalability challenges due to the computational inefficiency of estimating Jacobians. As a result, they typically scale to only dozens or hundreds of concepts [58], while interpretability in LLMs demands efficient modeling of thousands or even tens of thousands of concept features [54]. In summary, although CRL offers strong theoretical guarantees for recovering meaningful features and their causal relationships, its limited scalability in high-dimensional settings remains a major obstacle to practical deployment in LLM analysis.

To bridge this gap, in this paper, we introduce a computationally efficient temporal causal representation learning framework specifically designed for the high-dimensional activation space in LLMs. Our approach builds upon recent advances in both sparse autoencoders for LLMs and causal representation learning for sequential data. The key contributions of our work are:

(1) We propose a simple yet effective framework that jointly models time-delayed causal relations between concepts and instantaneous constraints, providing a more comprehensive understanding of how information flows through LLMs.

(2) Leveraging sparsity principle, we establish theoretical guarantees for our approach, making the representations learned reliable and explainable.

(3) Grounded in the theoretical result, we design scalable and efficient algorithms tailored to the high-dimensional concept space of LLMs, significantly extending prior work in CRL.

(4) We validate our approach on synthetic datasets scaled to match real-world complexity and demonstrate its effectiveness when applied to activations from real LLMs.

2 Problem Setting

We begin by characterizing the generation process of LLM activations to establish interpretability guarantees. These activations—signals produced during inference—are widely assumed to be linearly generated from hidden concepts, consistent with sparse autoencoder (SAE) literature [3, 18]. However, existing formulations typically treat these concepts as independent, overlooking dependencies between them. In reality, earlier-token semantics often influence later tokens, and token generation depends jointly on the activation of multiple concepts. To account for these interactions, we introduce a data generation process with both temporal and instantaneous relations, adopting CRL terminology. Given a token sequence $\boldsymbol{s} = (v_1, \ldots, v_k)$, let $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,m})$ be the n-dimensional activation vector at token v_t for a specific layer. Following the linear representation hypothesis [44] and SAE formulation [3, 18], we assume:

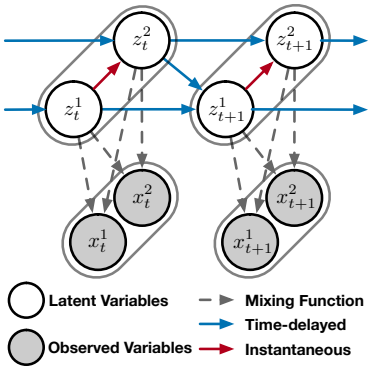


Figure 1: Graphical illustration of the data generation process.

$$\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t), \tag{1}$$

where $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ is the linear mixing function, and $\mathbf{x}_t$ and $\mathbf{z}_t$ are observed and latent variables, respectively. Besides, each latent variable $z_{t,i}$ is governed by a structural equation model (SEM) capturing both time-delayed and instantaneous dependencies:

$$z_{t,i} = \sum_{\tau} \sum_{j \in \mathcal{J}_{i,\tau}} B_{i,j,\tau} z_{t-\tau,j} + \sum_{j \in \mathcal{K}_i} M_{i,j} z_{t,j} + \epsilon_{t,i}, \tag{2}$$

where $B_{i,j,\tau}$ represents the coefficient for the time-delayed effect from $z_{t-\tau,j}$ to $z_{t,i}$; $\mathcal{J}_{i,\tau}$ is the set of indices of latent variables that have a time-delayed effect on $z_{t,i}$ with lag τ; $M_{i,j}$ represents the

coefficient for the instantaneous effect from $z_{t,j}$ to $z_{t,i}$; $\mathcal{K}_i$ is the set of indices of latent variables that have an instantaneous effect on $z_{t,i}$; and $\epsilon_{t,i}$ denotes the temporally and spatially independent noise extracted from a distribution p_{ϵ_i}. The graphical model for this process is illustrated in Figure 1.

To better understand this data generation process in the context of LLM activations, $\mathbf{x}_t$ represents activations in a specific layer l for token v_t, and the latent variables $\mathbf{z}_t$ can be considered as the underlying causal factors that generate these activations. In this case, the instantaneous effects (coefficients $M_{i,j}$) reflect semantic or syntactic relationships between different latent factors within the same token, while time-delayed effects (coefficients $B_{i,j,\tau}$) represent dependencies on previous tokens. Putting them together, the underlying data generating process can be written as

$$\underbrace{\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t),}_{\text{Linear mixing}} \quad \underbrace{z_{t,i} = \sum_{\tau>0} \sum_{j \in \mathcal{J}_{i,\tau}} B_{i,j,\tau} z_{t-\tau,j} + \sum_{j \in \mathcal{K}_i} M_{i,j} z_{t,j} + \epsilon_{t,i},\ i = 1, \ldots, m\,.}_{\text{Linear latent temporal SEM}} \tag{3}$$

The linear latent temporal SEM in Eq. (3) induces two types of causal relationships:

- A time-delayed causal graph $\mathcal{G}_d$ with vertices $\{z_{t,i}\}_{i=1}^n$ across different token positions and edges $z_{t-\tau,j} \to z_{t,i}$ if and only if $B_{i,j,\tau} \neq 0$.
- An instantaneous causal graph $\mathcal{G}_e$ with vertices $\{z_{t,i}\}_{i=1}^n$ at each token position t and edges $z_{t,j} \to z_{t,i}$ if and only if $M_{i,j} \neq 0$.

We assume that $\mathcal{G}_e$ is acyclic, i.e., a directed acyclic graph (DAG). This implies that the matrix M can be permuted to be strictly lower triangular, and the conditional distribution of variables $\mathbf{z}_t$ given their past values satisfy the Markov property w.r.t. DAG $\mathcal{G}_e$ [46], i.e., $p(\mathbf{z}_t|\mathbf{z}_{<t}) = \prod_{i=1}^n p(z_{t,i} \,|\, \mathrm{Pa}_d(z_{t,i}), \mathrm{Pa}_e(z_{t,i}))$.

Remark on the Linearity of the Model We acknowledge that the internal mechanisms of LLMs are inherently nonlinear due to activation functions and attention mechanisms. However, our linear approach is justified by several considerations. First, many successful mechanistic interpretability techniques [15, 43, 10, 1, 31] rely on linear representation hypotheses as approximations of localized network behavior. Second, linear models provide an interpretable bridge between the complexity of neural networks and human understanding—they serve as simplified yet informative projections of the underlying causal mechanisms. Third, empirical evidence suggests that linear approximations can capture significant portions of variance in activations within specific contexts [38, 19], particularly when examining feature-to-feature relationships within a layer.

More importantly, existing causal representation learning (CRL) methods cannot efficiently handle hundreds of latent variables, often encountering out-of-memory issues and prohibitively long computation times. A detailed discussion of these limitations is presented in Section 5.1. While nonlinear interactions certainly exist, our linear framework offers a tractable foundation for identifying causal relationships that can later be extended to incorporate more complex dependencies. This approach follows the scientific principle of starting with simpler models that capture essential phenomena before introducing additional complexity.

3 Theoretical Guarantees

Recent work in causal representation learning, especially for time-series data, has advanced to handle both time-delayed and instantaneous causal relations. Under general assumptions about the data generation process, strong identifiability results can be established. These include recovering latent variables up to component-wise transformations and estimating the Markov network up to isomorphic equivalence. In this section, we first introduce the definition of observational equivalence and identifiability. Then, we present the identifiability result from [28], which is established in a general non-linear setting. We then introduce a stronger identifiability result for the linear data generation process described in Eq. (3).

Definition 1 (Linear Modification from [59, 58])**.** *Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ be a sequence of observed variables generated by the true latent causal processes specified by $(A, B, p_{\boldsymbol{\epsilon}})$ given in Eq.* (3)*. A learned generative model $(\hat{A}, \hat{B}, \hat{p}_{\boldsymbol{\epsilon}})$ is observationally equivalent to $(A, B, p_{\boldsymbol{\epsilon}})$ if the model distribution $p_{\hat{A},\hat{B},\hat{p}_{\boldsymbol{\epsilon}}}(\{\mathbf{x}\}_{t=1}^T)$ matches the data distribution $p_{A,B,p(\boldsymbol{\epsilon})}(\{\mathbf{x}\}_{t=1}^T)$ for any value of*

$\{\mathbf{x}\}_{t=1}^{T}$. *We say latent causal processes are identifiable if observational equivalence can lead to a version of the generative process up to a permutation and scaling (linear component-wise transformation):*

$$p_{\hat{A},\hat{B},\hat{p}_{\boldsymbol{\epsilon}}}(\{\mathbf{x}\}_{t=1}^{T}) = p_{A,B,p_{\boldsymbol{\epsilon}}}(\{\mathbf{x}\}_{t=1}^{T}) \Rightarrow \hat{A} = APD, \tag{4}$$

where P is a permutation matrix and D is a diagonal invertible matrix.

Definition 2 (Intimate Neighbor Set [60])**.** *Consider a Markov network $\mathcal{M}_Z$ over variables set Z, and the intimate neighbor set of variable Z_i is*

$$\Psi_{M_Z}(Z_i) \triangleq \{Z_j \mid Z_j \text{ is adjacent to } Z_i \text{ and is also adjacent to all other neighbors of } Z_i, Z_j \in Z \setminus \{Z_i\}\}$$

Based on the aforementioned definitions, [28] showed that the latent variables can be identified under mild assumptions:

Theorem 1 ([28, Theorem 2])**.** *Suppose that the observations are generated by an instantaneous latent process, and $\mathcal{M}_{\mathbf{c}_t}$ is the Markov network over $\mathbf{c}_t = \{\mathbf{z}_{t-1}, \mathbf{z}_t, \mathbf{z}_{t+1}\} \in \mathbb{R}^{3n}$. Assume:*

- *A1 (Smooth and Positive Density). The conditional density $p(\mathbf{c}_t \mid \mathbf{z}_{t-2})$ is third-order differentiable and strictly positive on $\mathbb{R}^{3n}$.*
- *A2 (Sufficient Variability). With $|\mathcal{M}_{\mathbf{c}_t}|$ denoting the number of edges in $\mathcal{M}_{\mathbf{c}_t}$, define*

$$\begin{aligned} w(m) = &\left(\frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-2})}{\partial c_{t,1}^2 \partial z_{t-2,m}}, \ldots, \frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-2})}{\partial c_{t,2n}^2 \partial z_{t-2,m}}\right) \\ &\oplus \left(\frac{\partial^2 \log p(\mathbf{c}_t|\mathbf{z}_{t-2})}{\partial c_{t,1} \partial z_{t-2,m}}, \ldots, \frac{\partial^2 \log p(\mathbf{c}_t|\mathbf{z}_{t-2})}{\partial c_{t,2n} \partial z_{t-2,m}}\right) \oplus \left(\frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-2})}{\partial c_{t,i} \partial c_{t,j} \partial z_{t-2,m}}\right)_{(i,j)\in\mathcal{E}(\mathcal{M}_{\mathbf{c}_t})} \end{aligned} \tag{5}$$

  where $\oplus$ denotes concatenation operation and $(i,j) \in \mathcal{E}(\mathcal{M}_{\mathbf{c}_t})$ denotes all pairwise indice such that $c_{t,i}, c_{t,j}$ are adjacent in $\mathcal{M}_{\mathbf{c}_t}$. For every $m \in \{1, \ldots, n\}$ there exist $4n + |\mathcal{M}_{\mathbf{c}_t}|$ distinct values of $z_{t-2,m}$ such that the resulting $w(m)$ vectors are linearly independent.
- *A3 (Sparse Latent Process). For any $z_{it} \in \mathbf{z}_t$, the intimate neighbor set of $z_{t,i}$ is an empty set.*

When the observational equivalence is achieved with the minimal number of edges of the estimated Markov network of $\mathcal{M}_{\hat{\mathbf{c}}_t}$, then we have the following two statements:

(i) The estimated Markov network $\mathcal{M}_{\hat{\mathbf{c}}_t}$ is isomorphic to the ground-truth Markov network $\mathcal{M}_{\mathbf{c}_t}$.

(ii) There exists a permutation π of the estimated latent variables and a component-wise transformation $\mathcal{T}$, such that $z_{it} = \mathcal{T}(\hat{z}_{\pi(i)t})$, i.e., z_{it} is component-wise identifiable.

Implication When the underlying structure of latent variables exhibits sparsity (Assumption A3), the theorem guarantees that these variables can be uniquely identified up to permutation and component-wise transformation. This theoretical foundation aligns with and validates a key assumption in the LLM interpretability community: that meaningful concepts in large language models are characterized by sparse relations and influences.

When we identify latent concepts, the causal relations among them can also be ensured uniquely.

Theorem 2 ([28, Theorem 3])**.** *Suppose that the observations are generated following Theorem 1, and that $\mathcal{M}$ is the Markov network over two consecutive latent variables $\{\mathbf{z}_{t-1}, \mathbf{z}_t\} \in \mathbb{R}^{2n}$. Suppose that all assumptions for Theorem 1 hold. We further make the following assumption:*

- *A4 (Non-identical Parents) For any pair of adjacent latent variables $z_{t,i}, z_{t,j}$ at time step t, their time-delayed parents are not identical, i.e., $Pa_d(z_{t,i}) \neq Pa_d(z_{t,j})$.*

Then, the causal graph of the latent causal process is identifiable.

Implication: This theorem tells us as long as temporal patterns are unique for different concepts, such pattern can also be recovered.

Now we already have the theoretical guarantee to recover both latent variables together with their causal relations, further take the data generation process described in Eq. (3), we introduce a stronger version of identifiability in linear case bellow:

Proposition 1 (Identifiability of Latent Variables with Linear Temporal and Instantaneous Relations)**.** *Suppose that the observations are generated with temporal and instantaneous latent process as described in Eq.* (3) *with* $\tau = 1$*, and* $\mathcal{M}_{\mathbf{c}_t}$ *is the Markov network over* $\mathbf{c}_t = \{\mathbf{z}_{t-1}, \mathbf{z}_t, \mathbf{z}_{t+1}\} \in \mathbb{R}^{3n}$*. Assume the (A1) Smooth and positive density, (A2) Sufficient variability and (A3) Sparse Latent Process assumptions in Theorem 1 hold true, and the observational equivalence is achieved with the minimal number of edges of the estimated Markov network of* $\mathcal{M}_{\hat{\mathbf{c}}_t}$*, then*

(i) The estimated Markov network $\mathcal{M}_{\hat{\mathbf{c}}_t}$ *is isomorphic to the ground-truth Markov network* $\mathcal{M}_{\mathbf{c}_t}$*.*

(ii) There exists a permutation matrix P *and a scaling matrix* D *of the estimated latent variables, such that* $\mathbf{z}_t$ *is identifiable up to permutation and scaling, i.e.,* $\mathbf{z}_t = PD\,\hat{\mathbf{z}}_t$*.*

Further assume the (A4) Non-identical time-delayed parents condition in Theorem 2, then the causal graph of the latent causal process is identifiable.

Proof Sketch The proof is straightforward by leveraging the result from Theorem 1, 2 and applying the linearity of the data generation process. In particular, given the linear mixing function A, the component-wise transformation must be linear, which leads to the final results. And the multi-lag setting ($\tau > 1$) can also be handled with extensions described in Appendix B.2 in [28]. We defer the complete proof and detailed extensions to the multi-lag setting to Appendix A.1.

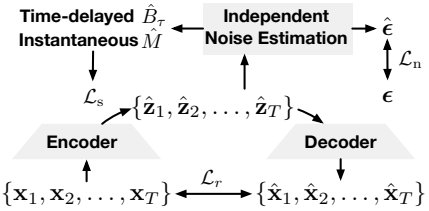


Figure 2: Illustration of estimation process. $\hat{B}_\tau$ represents the learned time-delayed causal relation and $\hat{M}$ is the instantaneous causal relation.

4 Implementation

Based on the data generation process in Eq. (3) together with the identifiability result presented in Theorem 1, we derive the following estimation process based on the standard sparse autoencoder. Illustrated in Figure 2, the whole estimation process can be partitioned into three parts, namely (1) observation reconstruction, (2) independent noise estimation, and (3) sparsity regularization.

4.1 Observation Reconstruction

First, we use a linear autoencoder to enforce the invertible linear transformation between observations $\mathbf{x}_t$ and latent variables $\mathbf{z}_t$, and the reconstruction loss $\mathcal{L}_r$ is defined as

$$\mathcal{L}_r = \mathbb{E}_{\mathbf{x}_{1:T}} \left[\sum_{t=1}^{T} (\mathbf{x}_t - \hat{\mathbf{x}}_t)^2 \right], \tag{6}$$

where the reconstructed observation is calculated via a linear encoder and decoder:

$$\hat{\mathbf{x}}_t = \texttt{Decoder}(\hat{\mathbf{z}}_t) \quad \text{and} \quad \hat{\mathbf{z}}_t = \texttt{Encoder}(\mathbf{x}_t). \tag{7}$$

4.2 Independent Noise Estimation

In prior works [58, 50, 28], this terms refers to the independent prior estimation, in which they essentially utilize the independence of noise to enforce the independence of latent variables $z_{t,i}$, conditioning on parent $\mathrm{Pa}(z_{t,i})$. In our case, since the whole process is linear, we can directly estimate and enforce the independent noise condition by learning a residual network by reversing the data generation process described in Eq. 3:

$$\hat{\boldsymbol{\epsilon}}_t = \hat{\mathbf{z}}_t - \hat{M}\,\hat{\mathbf{z}}_t - \sum_{\tau>0} \hat{B}_\tau\,\hat{\mathbf{z}}_{t-\tau}, \tag{8}$$

where the estimated latent variables are given by Eq. (7). Following the prior works, to enforce the independence of noise terms, we model the noise distribution $p(\hat{\epsilon}_{t,i})$ with isomorphic Laplacian[†] distribution, and we minimize its KL-divergence with the estimated noise term.

$$\mathcal{L}_n = \mathbb{E}_{\hat{\boldsymbol{\epsilon}}_t} \left[||\hat{\boldsymbol{\epsilon}}_t||_1 \right]. \tag{9}$$

[†]In prior works, the distribution is Gaussian, however, we can see that in linear case as is well discussed in linear ICA literature, the density function of an isomorphic Gaussian distribution is rotation invariant, hence we utilize the Laplacian distribution in our estimation.

4.3 Sparsity Regularization

Without any further constraint, the noise estimation module may bring redundant causal edges from $\hat{\mathbf{z}}_{t-1}, \hat{\mathbf{z}}_{t,[m]\setminus i}$ to $\hat{z}_{t,i}$, leading to the incorrect estimation. As mentioned in Sec. 4.2, $\{B_\tau\}$ and M intuitively denote the time-delayed and instantaneous causal structures, since they describe how the $\hat{\mathbf{z}}_{t-1}, \hat{\mathbf{z}}_{t,[m]\setminus i}$ contribute to $\hat{z}_{t,i}$, which motivate us to remove these redundant causal edges with a sparsity regularization term $\mathcal{L}_s$ by using the L1 penalty on $\{\hat{B}_\tau\}$ and $\hat{M}$. Formally, we have

$$\mathcal{L}_s = \left(\sum_{\tau} ||\hat{B}_\tau||_1 \right) + ||\hat{M}||_1, \tag{10}$$

where $|| * ||_1$ denotes the L1 Norm of a matrix. By employing the gradient-based sparsity penalty on the estimated latent causal processes, we can indirectly restrict the sparsity of Markov networks to satisfy the sparse latent process. Finally, the total loss of the model can be formalized as:

$$\mathcal{L}_{total} = \mathcal{L}_r + \alpha \mathcal{L}_n + \beta \mathcal{L}_s, \tag{11}$$

where α, β denote the hyper-parameters.

5 Experiments

Our experimental evaluation addresses five key claims regarding our proposed method: (1) our estimation approach aligns with identifiability theory, accurately recovering latent structures; (2) existing CRL methods fail to handle high-dimensional data at scale; (3) our method is able to recover target relations between concepts from semi-synthetic data; (4) compared with common SAEs, our proposal achieves satisfactory results on quantitative evaluation metrics (SAEBench [25]); and (5) our method effectively learns both time-delayed and instantaneous causal relations among concepts elicited from LLM activations.

5.1 Synthetic Data Experiments

First, using synthetic data, we demonstrate that our method can recover both the latent variables $\mathbf{z}_t$ and the causal structure including time-delayed relations B_τ and instantaneous relations M.

Identifiability Verification To establish the effectiveness of our approach, we generate simulated time series data with a latent causal process as introduced in Eq. (3). We apply our method to single time lag synthetic data generated with a randomly initialized matrix A and fixed transition matrices B and M visualized in Figure 3a and 3c. Further details can be found in Appendix A.2.

We visualize the estimated parameters by plotting the recovered matrices $\hat{B}$ and $\hat{M}$ alongside the correlation coefficient matrix used for calculating the mean correlation coefficient (MCC) score.

As shown in Figure 3, comparing with the ground truth transition matrices B and M, we observe that both time-delayed and instantaneous causal relations have been precisely recovered. Furthermore, Figure 3e demonstrates that the latent variables $\mathbf{z}_t$ are also accurately recovered, confirming the identifiability properties of our method.

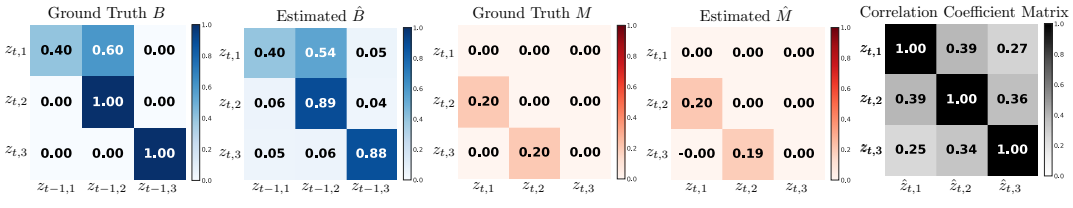
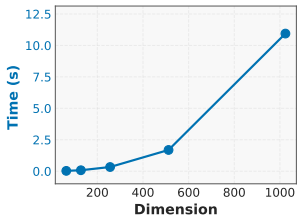


Figure 3: Visualization of recovered causal graphs of latent variables. (a) and (b) show the ground truth and estimated time-delayed matrix, respectively. (c) and (d) show the ground truth and the estimated instantaneous causal relations, respectively. (e) displays the correlation between the ground truth and recovered latent variables.

Second, we scale the synthetic experiments to dimensions matching LLM activations, illustrating why existing CRL methods fail in these high-dimensional settings.

Challenges on Scaling to Large Language Model Activation Dimensions Before presenting results on expanded synthetic data, we investigate the computation bottleneck: Jacobian calculation and explain why existing CRL methods do not extend efficiently to high-dimensional settings, thereby further motivating our use of a linear dynamical model.



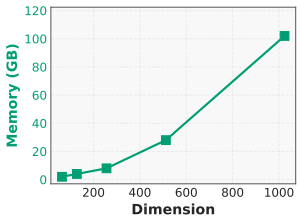


Figure 5: Computation time and memory usage for a single-step Jacobian as a function of input dimensionality. Both metrics grow superlinearly and exceed the capacity of modern GPUs when the input dimension is greater than 1000.

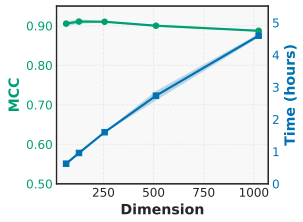


Figure 6: MCC and total compute time in hours required to train the linear model as a function of input dimension.

Computation cost of Jacobian evaluation. We take IDOL [28] as a representative method and measure both the wall-clock time and memory requirements for computing the Jacobian in prior network. Figure 5 demonstrates that both time and memory complexity grow polynomial with dimensionality. At dimensions of several thousand—which are common for Large Language Model activations—a single Jacobian evaluation will require approximately ten seconds on a modern GPU, such computation cannot fit into current generation hardware infrastructure. Since CRL training invokes this operation millions of times during the training process, the cumulative computational cost becomes prohibitive. As other CRL algorithms involve comparable Jacobian computations or more complex algorithms, this fundamental limitation applies broadly across the field.

Advantages of linear models. When a linear model provides an adequate approximation of the transition dynamics of the hidden concepts, the Jacobian calculation can be derived directly from model parameters such as B and M, which significantly reduces the computational burden. Furthermore, such a linear model can scale efficiently with current-generation compute resources. To support this claim, we conducted a scaling experiment using the linear model on synthetic data with dimensionalities ranging from 128 to 1024. In each setting, the model was trained on 50 million samples, simulating the typical training load in real LLM SAEs with 50 million tokens. As shown in Figure 6, the proposed method scales to substantially higher dimensions while maintaining a high MCC of approximately 0.9. Additionally, it remains computationally efficient, with total computation time scaling linearly. In contrast, IDOL [28] exhausts memory when the dimensionality exceeds 200, and iCITRIS [32] fails to scale beyond 16 dimensions.

5.2 Semi-synthetic Experiments

Given the previous experiments on synthetic data, our proposal has been shown to recover ground-truth relationships even when the hidden dimensionality reaches one thousand, which would be challenging for existing non-parametric CRL approaches. We now proceed to evaluate real-world LLM activations, beginning with investigation (3). The experimental settings are briefly introduced below. Full details can be found in Appendix A.3.1.

Table 1: Relation recovery scores (↑) for concept–relation extraction on semi-synthetic data.

Method	Legal	XML	Email
SAE+regression	0.54	0.94	0.74
Ours	**19.95**	**8.63**	**2.66**

Data preparation: We first examine three types of text, each exhibiting an obvious syntactic pattern. For example, in legal text, sequences often begin with "APPEALS" and end with "AFFIRMED". For illustration, we focus on the legal text contrastive corpus group. We constructed two contrastive subsets from the `Pile` dataset [17]: one containing legal documents with highly structured syntax and stable temporal patterns, and the other containing unstructured non-legal text. We hypothesized that

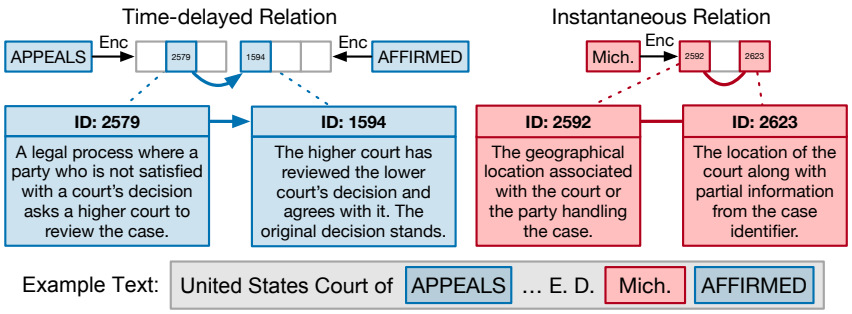


Figure 7: Case study illustrates two relation types identified in a United States legal text. The blue elements show a time-delayed relation: the term "appeals" is typically followed by "affirmed" when a higher court confirms the lower court decision. The red elements show an instantaneous relation: two geographical location concepts (2592, 2623), are activated together in the same passage.

only texts containing these structured relations would yield meaningful temporal concept patterns and tested whether the model could recover them. **Baseline:** Since no directly applicable baseline exists, we used the standard SAEs trained above. As SAEs cannot capture concept-to-concept relations, we fitted a regression model to estimate temporal relation matrices $\tilde{B}$ via $\mathbf{z}_t = \sum_\tau \tilde{B}_\tau \mathbf{z}_{t-\tau}$. **Evaluation:** We compute the concept recovery score by first identifying the top concept pair (i, j) in legal contexts (ensuring that the two corresponding concepts do not fire in the non-legal text), then taking the corresponding coefficient $B_{i,j}$ and normalizing it by the standard deviation $\sigma(B)$. The ratio $\frac{B_{i,j}}{\sigma(B)}$ serves as a relation recovery score, indicating how strongly the relation stands out from noise. As shown in Table 1, our method achieves a significantly higher score, demonstrating successful recovery of the relation. Finally, as concept recovery is already achievable by standard SAEs, we additionally conducted steering vector semi-synthetic experiments to verify that our proposal can also recover concepts, following the approach of [24]. Further details are provided in Appendix A.3.2.

5.3 Real LLM Activation Analysis

Experiment Setup We train our linear model on activations from the pretrained LLM `pythia-160m-deduped` [5], using `SAELens` [6] and `dictionary-learning` [36] for activation extraction. The model is trained on 50 million tokens from the `Pile` dataset [17]. To capture time-delayed influences, we set $\tau \leq 20$ in Eq. 3 and aggregate the B_τ matrices using max-pooling, preserving any causal link that appears at any time step. We evaluate three feature dimensionalities: 768 (matching the LLM's hidden size and aligned with Section 3), 3072, and 6144—the latter two following common SAE training setups. Unless specified, main text examples use 3072-dimensional features with $\tau = 20$. Full training details and additional results (sensitivity and ablation studies are included) are in Appendix A.4.

Table 2: Comparison of our method against ReLU and TopK SAEs on SAEBench metrics.

Model	Recon. Loss ↓	Sparse Prob. ↑	Absorp. ↓	Autointerp ↑
ReLU SAE	0.0110	0.6555	0.0141	0.6791
TopK SAE	**0.0097**	**0.7141**	0.0280	0.6822
Ours	0.0108	0.6736	**0.0139**	**0.6883**

Quantitative Evaluation on SAEBench Before we dive into the details of concept relation recovery, we first present a quantitative comparison between our method and existing SAE approaches. Since our main contribution lies in recovering temporal and instantaneous concept-to-concept relations, which are not reflected in current SAE benchmarks, we expect our model to perform on par with established SAEs on SAEBench tasks. This expectation is confirmed by the results in Table 2. Additional experiment results on larger latent size and model size can be found in Appendix A.4.5.

Case Studies We start with an illustrative case in Figure 7, demonstrating how our model uncovers interpretable concept features with both time-delayed and instantaneous causal relationships from real-world LLM activations. This example provides an integrated view of how concepts are structured over time and interact within a single time step. Note that feature interpretations may vary beyond this case; additional examples and discussions appear in Appendix A.4.

Table 3: Representative time-delayed and instantaneous concept relations discovered.

ID	From	ID	To	Coeff.
Time-delayed relations				
1657	Keywords for formal and official content (e.g., *senate*, *state*, *military*)	1664	Verbs for official/formal usage (e.g., *deny*, *press*, *order*, *sign*)	0.99
2641	Adjectives of nationality (e.g. *Japanese*, *Italians*)	2674	Nouns that follow nationality (e.g. *brands*, *name*)	0.81
1657	Keywords for formal and official content (e.g., *senate*, *state*, *military*)	1124	Objective adjectives in formal usage (e.g., *fast*, *continuous*, *incomplete*)	0.74
Instantaneous relations				
2208	Partial appellate citation with volume number	227	Partial appellate citation with volume number and series index	0.23
1714	Coding-format signals (e.g. *localization tags*, *HTML* tags)	80	Coding-format content (e.g. *key–value pairs*, *HTML elements*)	0.16
1582	Month (e.g. *March*)	363	Full date (e.g. *March 23, 2000*)	0.02

Figure 7 highlights two key observations: First, a time-delayed causal link between concepts related to "appeals" and "affirmed" in legal texts (features 2579 and 1594), capturing how the model reflects the procedural flow of legal judgments. Second, an instantaneous relation between two geographical location concepts (features 2592 and 2623) that are activated together in legal passages, suggesting that the model represents related spatial information simultaneously rather than sequentially. This example effectively demonstrates that both time-delayed and instantaneous relations exist among concept features, and that these are interpretable alongside the semantic meanings of the features—both of which are essential for LLM interpretability.

To further demonstrate our model's capacity to uncover both types of causal relationships, we present a broader set of examples in Table 3, which showcases representative cases of both time-delayed and instantaneous interactions among concept features.

Time-delayed causal relations. We first observe a strong causal relation from nationality adjectives (feature 2641, "Japanese," "Italians") to the nouns they commonly modify (feature 2674, "brands," "literature"), with a coefficient of 0.81. Moreover, the coefficients across the 20-token temporal window (i.e., different B_τ) contribute consistently to the aggregated score. This suggests that such temporal relations can occur across a broad and uncertain time span, aligning with the semantic dynamics of real-world text generation. In formal contexts, official content words (feature 1657, "senate," "judge") influence both formal verbs (feature 1664, "deny," "order") with a coefficient of 0.99, and objective adjectives (feature 1124, "fast," "continuous") with a coefficient of 0.74. These relationships reflect how formal language constrains both action and descriptive style over time.

Instantaneous causal relations. Table 3 presents three distinct categories of instantaneous relations. First, we observe a relationship between two partially overlapping appellate citation features—feature 2208 (volume numbers only) and feature 227 (volume number and series index)—with a normalized coefficient of 0.23. This illustrates how the model captures structured elements that commonly co-occur in legal documents, forming a cohesive representational unit. Second, we find that coding-format signals (feature 1714, e.g., localization tags, HTML tags) have an instantaneous causal relationship with coding-format content (feature 80, e.g., key-value pairs, HTML elements), with a coefficient of 0.16. This reveals how the model processes structured syntax and its associated content as co-occurring elements. Finally, our method identifies a clear relationship between two features that both represent dates: feature 1582 (month only) and feature 363 (full date), suggesting complementary representations within the model's internal structure.

These findings demonstrate our method's ability to uncover both temporal and instantaneous causal structures in the concept space of LLM activations, offering insights into how models organize and process information. The identified relationships align with expected patterns in natural language across domains such as legal texts, temporal expressions, and structured formats, validating the effectiveness of our approach for analyzing information flow in large language models.

6 Related Work

LLM Interpretability Understanding the internal representations of LLMs remains challenging despite significant progress [27]. Interpretability research on LLMs has explored multiple directions including: probing for linguistic knowledge [19], evaluating interpretability methods through controlled experiments [23], benchmarking SAEs' capacity to disentangle factual knowledge [12], and developing ground-truth evaluation frameworks [55, 26]. Recent work suggests that LLM representations may follow a linear organization [44], though this hypothesis has been challenged [16]. Our approach extends these efforts by focusing specifically on causal interpretability of temporal relationships in LLMs, providing a principled framework for understanding how information flows through model representations during sequential text generation. Additionally sparse autoencoders (SAEs) decompose neural activations into interpretable features [14, 7, 54]. Initial work demonstrated that SAEs can recover meaningful features from language model activations [42], leading to numerous architectural innovations including alternative activation functions [52, 47], training optimizations [8, 20], and efficient dictionary allocation mechanisms [4, 40]. Recent work has successfully scaled SAEs to larger models [18, 30, 2], enabling automated interpretation of millions of features [45]. Despite these advances, most SAE approaches treat features as isolated units without modeling temporal relationships [11, 9], lack explicit causal structure [35], and offer no identifiability guarantees [56, 34]—limitations our work directly addresses.

Feature-based Causal Circuits Recent methods like Sparse Feature Circuits [37] and attribution graphs [1, 31] identify causal subnetworks explaining model behavior. These build on earlier circuit analysis methods exploring component functionalities in vision and language models [43, 10, 15]. Targeted interventional studies have revealed specific functional circuits, such as indirect object identification [57] and factual associations [38]. While these methods enable mechanistic understanding of model computations [19, 41], they primarily rely on correlational measures rather than structured causal inference [16, 44]. But they focus on stationary relationships [21] instead of modeling evolving token-to-token dependencies critical for understanding sequential reasoning.

Causal Representation Learning Causal representation learning provides identifiability guarantees for latent variables [58, 28, 48]. Temporal extensions model dynamics in sequential data [59, 32, 50], with recent advances addressing non-stationarity [49, 13] and instantaneous effects [33]. Multiple distribution methods [60, 39] can recover causal structure under specific interventions or group structures. These approaches provide theoretical foundations for disentangling latent variables and identifying causal graphs [56]. However, existing CRL algorithms cannot scale to LLM dimensions due to computational bottlenecks in calculating Jacobians. Our linearized formulation maintains identifiability guarantees while enabling application to high-dimensional LLM representations—bridging theoretical CRL advances with practical LLM interpretability challenges.

7 Conclusion

We introduced a causal representation learning framework for LLMs that jointly models time-delayed relationships and instantaneous constraints between latent concepts. Our approach provides theoretical identifiability guarantees while solving the scalability limitations of existing CRL methods through a computationally efficient linear formulation. Synthetic experiments validated our method's ability to recover latent causal structures from toy scale to real LLM scales. When applied to real LLM activations, our approach uncovered interpretable semantic patterns, revealing information flow pathways during text generation. Future work could leverage these causal structures for targeted alignment interventions, explore cross-layer concept transformations, and integrate with mechanistic interpretability techniques.

8 Acknowledgment

The authors would like to thank the anonymous reviewers for helpful comments and suggestions during the reviewing process. The authors would also like to acknowledge the support from NSF Award No. 2229881, AI Institute for Societal Decision Making (AI-SDM), the National Institutes of Health (NIH) under Contract R01HL159805, and grants from Quris AI, Florin Court Capital, and MBZUAI-WIS Joint Program, and the Al Deira Causal Education project.

References

[1] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025.

[2] Anthropic Interpretability Team. Circuits updates — august 2024. *Transformer Circuits Thread*, 2024.

[3] Anthropic Interpretability Team. Training sparse autoencoders. `https://transformer-circuits.pub/2024/april-update/index.html#training-saes`, 2024. [Accessed January 20, 2025].

[4] Kola Ayonrinde. Adaptive sparse allocation with mutual choice & feature choice sparse autoencoders, 2024.

[5] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward 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.

[6] Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. Saelens. `https://github.com/jbloomAus/SAELens`, 2024.

[7] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[8] Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk: A simple improvement for topk-saes, 2024.

[9] Bart Bussmann, Michael Pearce, Patrick Leask, Joseph Isaac Bloom, Lee Sharkey, and Neel Nanda. Showing sae latents are not atomic using meta-saes, 2024.

[10] Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 6(1):e00024–006, 2021.

[11] David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2024.

[12] Maheep Chaudhary and Atticus Geiger. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small, 2024.

[13] Guangyi Chen, Yifan Shen, Zhenhao Chen, Xiangchen Song, Yuewen Sun, Weiran Yao, Xiao Liu, and Kun Zhang. Caring: Learning temporal causal representation under non-invertible generation process. *arXiv preprint arXiv:2401.14535*, 2024.

[14] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023.

[15] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

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

[17] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[18] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[19] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[20] Davide Ghilardi, Federico Belotti, and Marco Molinari. Efficient training of sparse autoencoders for large language models via layer groups. *arXiv preprint arXiv:2410.21508*, 2024.

[21] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023.

[22] Christopher J Hillar and Friedrich T Sommer. When can dictionary learning uniquely recover sparse data from subsamples? *IEEE Transactions on Information Theory*, 61(11):6290–6297, 2015.

[23] Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations, 2024.

[24] Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. Identifiable steering via sparse autoencoding of multi-concept shifts. *arXiv preprint arXiv:2502.12179*, 2025.

[25] Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, et al. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*, 2025.

[26] Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models, 2024.

[27] Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2023.

[28] Zijian Li, Yifan Shen, Kaitao Zheng, Ruichu Cai, Xiangchen Song, Mingming Gong, Guangyi Chen, and Kun Zhang. On the identification of temporal causal representation with instantaneous dependence. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

[30] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024.

[31] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025.

[32] Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves. Causal representation learning for instantaneous and temporal effects in interactive systems. In *The Eleventh International Conference on Learning Representations*, 2023.

[33] Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Stratis Gavves. Citris: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, pages 13557–13603. PMLR, 2022.

[34] Aleksandar Makelov, George Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control, 2024.

[35] Luke Marks, Alasdair Paren, David Krueger, and Fazl Barez. Enhancing neural network interpretability with feature-aligned sparse autoencoders, 2024.

[36] Samuel Marks, Adam Karvonen, and Aaron Mueller. dictionary_learning. `https://github.com/saprmarks/dictionary_learning`, 2024.

[37] Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2024.

[38] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

[39] Hiroshi Morioka and Aapo Hyvärinen. Causal representation learning made identifiable by grouping of observational variables. *arXiv preprint arXiv:2310.15709*, 2023.

[40] Anish Mudide, Joshua Engels, Eric J Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient dictionary learning with switch sparse autoencoders. *arXiv preprint arXiv:2410.08201*, 2024.

[41] Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, et al. The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability. *arXiv preprint arXiv:2408.01416*, 2024.

[42] Neel Nanda. Open Source Replication & Commentary on Anthropic's Dictionary Learning Paper, Oct 2023.

[43] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[44] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[45] Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models, 2024.

[46] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

[47] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

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

[49] Xiangchen Song, Zijian Li, Guangyi Chen, Yujia Zheng, Yewen Fan, Xinshuai Dong, and Kun Zhang. Causal temporal representation learning with nonstationary sparse transition. *Advances in Neural Information Processing Systems*, 37:77098–77131, 2024.

[50] Xiangchen Song, Weiran Yao, Yewen Fan, Xinshuai Dong, Guangyi Chen, Juan Carlos Niebles, Eric Xing, and Kun Zhang. Temporally disentangled representation learning under unknown nonstationarity. *Advances in Neural Information Processing Systems*, 36:8092–8113, 2023.

[51] Yuchen Sun and Kejun Huang. Global identifiability of overcomplete dictionary learning via l1 and volume minimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] Glen M. Taggart. Prolu: A nonlinearity for sparse autoencoders, 2024. `https://www.alignmentforum.org/posts/HEpufTdakGTTKgoYF/prolu-a-nonlinearity-for-sparse-autoencoders`.

[53] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[54] Andrew Templeton, Timothy Conerly, Jacob Marcus, John Lindsey, Tamera Bricken, Bowen Chen, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Technical report, Anthropic, 2024. Transformer Circuits Thread Technical Report.

[55] Constantin Venhoff, Anisoara Calinescu, Philip Torr, and Christian Schroeder de Witt. Sage: Scalable ground truth evaluations for large sparse autoencoders, 2024.

[56] Julius von Kügelgen. Identifiable causal representation learning: Unsupervised, multi-view, and multi-environment. *arXiv preprint arXiv:2406.13371*, 2024.

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

[58] Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. *arXiv preprint arXiv:2210.13647*, 2022.

[59] Weiran Yao, Yuewen Sun, Alex Ho, Changyin Sun, and Kun Zhang. Learning temporally causal latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*, 2021.

[60] Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from multiple distributions: a general setting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

A Technical Appendices and Supplementary Material

Contents

A.1 Proof for Theorem 3

Proof. We start from the result in Theorem 1, in particular (ii) There exists a permutation π of the estimated latent variables and a component-wise transformation $\mathcal{T}^{\ddagger}$, such that

$$z_{it} = \mathcal{T}(\hat{z}_{\pi(i)t}),$$

i.e., z_{it} is component-wise identifiable. Then consider the assumption that the mapping from latent concept $\mathbf{z}$ to observations $\mathbf{x}$ is linear, and the fact that in estimations, the estimated encoder is also assumed to be a linear function, that is saying the component-wise transformation mentioned in the result of Theorem 1 is restricted to linear transformation

$$\mathbf{z}_t = T(\hat{\mathbf{z}}_t),$$

where T is a square matrix we show in the following lemma to decompose the T into a permutation and a diaginal matrix.

Lemma 1. *Let T be a component-wise linear transformation, meaning that for every standard basis vector e_i the image $T(e_i)$ has at most one non-zero coordinate. Then there exist a permutation matrix P and a diagonal matrix D such that $T = PD$.*

[‡]Note that for the case when the dimension of $\mathbf{x}$ matches the dimension of $\mathbf{z}$, then the bijection assumption in [28] can be easily adapted into the linear case. Even if the dimension doesn't match, we can still use this framework because under the condition that the latent variables are sparse, in which is exactly the sparse autoencoder setting, it can still be viewed as an invertible transformation, such claimed has already been extensively studied in overcomplete sparse dictionary learning literature [22, 51].

Proof. Linearity ensures that T is determined by its action on the basis $\{e_1, \ldots, e_n\}$. For each index i there is a scalar $\alpha_i \in \mathbb{F}$ and an index $\sigma(i) \in \{1, \ldots, n\}$ satisfying

$$T(e_i) = \alpha_i \, e_{\sigma(i)};$$

this follows from the component-wise assumption.

Define the permutation σ by the rule above and form the permutation matrix

$$P = \big[e_{\sigma(1)} \ \cdots \ e_{\sigma(n)}\big].$$

Set $D = \operatorname{diag}(\alpha_1, \ldots, \alpha_n)$.

For every basis vector e_i we have

$$PD\, e_i = P\big(\alpha_i e_i\big) = \alpha_i P e_i = \alpha_i e_{\sigma(i)} = T(e_i).$$

Since PD and T coincide on the basis, they coincide on all of $\mathbb{F}^n$; hence $T = PD$.

To see uniqueness, suppose $T = P_1 D_1 = P_2 D_2$ with permutation matrices P_1, P_2 and diagonal matrices D_1, D_2. Then $P_2^{-1} P_1 = D_2 D_1^{-1}$ is simultaneously permutation and diagonal, forcing it to be the identity. Consequently $P_1 = P_2$ and $D_1 = D_2$. □

Then for a component-wise linear transformation, the only possible solution is a permutation and a diaginal matrix i.e.

$$T = P\, D,$$

where P is the permutation matrix and D is the diagnal matrix. Then the latent variables are identifiable up to permutation and scaling, i.e., $\mathbf{z}_t = PD\, \hat{\mathbf{z}}_t$. □

We also include the discussion for higher order generalization which is originally given by [28] as follows: For any given τ, and subsequence which is centered at $\mathbf{z}_t$ with previous lo and following hi steps, i.e., $\mathbf{c}_t = \{\mathbf{z}_{t-lo}, \cdots, \mathbf{z}_t, \cdots, \mathbf{z}_{t+hi}\} \in \mathbb{R}^{(lo+hi+1)\times n}$. In this case, the vector function $w(i, j, m)$ in Sufficient Variability Assumption should be modified as

$$\begin{aligned} w(i,j,m) = &\Big(\frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-lo-1}, \cdots, \mathbf{z}_{t-lo-\tau})}{\partial c_{t,1}^2 \partial z_{t-lo-1,m}}, \cdots, \frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-lo-1}, \cdots, \mathbf{z}_{t-lo-\tau})}{\partial c_{t,2n}^2 \partial z_{t-lo-1,m}}\Big) \oplus \\ &\Big(\frac{\partial^2 \log p(c_t|\mathbf{z}_{t-lo-1}, \cdots, \mathbf{z}_{t-lo-\tau})}{\partial c_{t,1} \partial z_{t-lo-1,m}}, \cdots, \frac{\partial^2 \log p(c_t|\mathbf{z}_{t-lo-1}, \cdots, \mathbf{z}_{t-lo-\tau})}{\partial c_{t,2n} \partial z_{t-lo-1,m}}\Big) \oplus \\ &\Big(\frac{\partial^3 \log p(\mathbf{c}_t|\mathbf{z}_{t-lo-1}, \cdots, \mathbf{z}_{t-lo-\tau})}{\partial c_{t,i} \partial c_{t,j} \partial z_{t-lo-1,m}}\Big)_{(i,j)\in\mathcal{E}(\mathcal{M}_{\mathbf{c}_t})}. \end{aligned} \tag{12}$$

Besides, $2 \times n \times (lo + hi + 1) + |\mathcal{M}_{\mathbf{c}_t}|$ values of linearly independent vector functions in $z_{t',m}$ for $t' \in [t - lo - 1, \cdots, t - lo - \tau]$ and $m \in [1, \cdots, n]$ are required as well. Since such modification doesn't require the non-linear property of the function then the rest part of the theorem remains the same, and the proof can be easily extended in such a setting.

A.2 Synthetic Experiments

We conduct two synthetic verification experiments to validate our linear temporal instantaneous ICA method. Instruction is provided in the `synthetic/README.md` file in our code repository.

A.2.1 Fixed Structure Experiment

For the first synthetic verification experiment, we generate data using fixed time-delayed influence functions and instantaneous relations with the following ground truth matrices:

$$B = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 0 & 0 & 0 \\ 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \end{bmatrix}. \tag{13}$$

The data generation process follows a structured temporal model. We initialize the first hidden state z_0 by sampling from a uniform distribution $\mathcal{U}(0, 1)$. For subsequent time steps, we compute the

historical influence as $\mathbf{z}_{\text{hist}} = B\,\mathbf{z}_{t-1}$ and then construct z_t iteratively: the first dimension receives only historical influence plus noise, while remaining dimensions $i \geq 2$ incorporate both historical and instantaneous dependencies:

$$z_t^{(1)} = z_{\text{hist}}^{(1)} + \epsilon_t^{(1)} \tag{14}$$

$$z_t^{(i)} = z_{\text{hist}}^{(i)} + w_{\text{inst}} \cdot z_t^{(i-1)} + \epsilon_t^{(i)}, \quad i \geq 2 \tag{15}$$

where ϵ_t is Laplace noise with scale 1.0, and $w_{\text{inst}} = 0.2$. The observations are generated as $\mathbf{x}_t = A\mathbf{z}_t$ where A is a 3×3 randomly initialized mixing matrix .

We train the model for 50,000 steps with batch size 1024 (approximately 51 million total samples) using the Adam optimizer with learning rate 8×10^{-3} and weight decay 6×10^{-4}. The loss function includes reconstruction error, KL divergence term, and L1 regularization penalties: 1×10^{-3} for matrix M and 1×10^{-8} for matrix B. We enforce the lower-triangular constraint on M to ensure identifiability.

A.2.2 Scalability Experiment

For the second synthetic experiment, we evaluate scalability across different dimensions ranging from 64 to 1024. We randomly sample a sparse time-delayed transition matrix B where only 10% of the entries are non-zero, generated using a randomly initialized matrix with 10% masking.

For the instantaneous mixing matrix M, we use a chain structure where $M_{i,i-1} = 0.5$ for $i \geq 2$ and all other entries are zero:

$$M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0.5 & 0 & 0 & \cdots & 0 \\ 0 & 0.5 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0.5 & 0 \end{bmatrix} \tag{16}$$

The training hyperparameters are modified from the first experiment: learning rate increased to 1×10^{-3} and the sparsity coefficient for B increased to 1×10^{-5} to account for the higher dimensional setting, while maintaining 1×10^{-3}.

Both experiments use identical noise characteristics (Laplace distribution with unit scale), sequence length of 1 (two time steps total), and Mean Correlation Coefficient (MCC) as the primary evaluation metric to measure the quality of source recovery while accounting for permutation ambiguity inherent in ICA methods.

A.3 Semi-synthetic Experiments

A.3.1 Target Concept Relation Recovery

Before attempting to recover concept relationships from real-world LLM activations, and based on the proven and verified identifiability of our model, we first present a semi-synthetic setting to verify that our proposal can reveal obvious concept relations from contrastive corpus pairs.

Data Preparation We constructed two contrastive collections of texts drawn from the `Pile` dataset [17]. We considered three types of text: legal documents, emails, and XML files. For each type, we constructed two contrastive corpora: one containing the relation of interest, and the other lacking it. Specifically, for legal text, the relation is defined by sequences beginning with "APPEALS" and ending with "AFFIRMED"; for emails, sequences start with forwarding or reply markers (e.g., dashes) and end with common words like "Subject" or "Thanks"; and for XML, sequences start with a UTF encoding label followed by tags such as "UTF-8" or "!DOCTYPE". We hypothesized that only texts containing these structured relations would yield meaningful temporal concept patterns, and we directly tested whether the model can successfully recover such patterns.

Baseline Construction Although there is no directly applicable baseline, we leveraged standard SAEs we had trained above to serve as our baseline method. Since SAEs themselves cannot capture the concept-to-concept relations, we train a regression model to find temporal relation coefficient matrices $\tilde{B}$s by solving the following regression task: $\mathbf{z}_t = \sum_\tau \tilde{B}_\tau \mathbf{z}_{t-\tau}$.

Evaluation Metric We calculate the concept recovery score by first obtaining the top fired feature index pair (i, j) related to the legal context (restricted to positions where the concepts of interest ought to fire but do not fire in the contrastive non-legal text), and then taking the corresponding entry $B_{i,j}$ in the aggregated temporal relation coefficient matrices. We then calculate the relation recovery score, similar to a signal-to-noise ratio, by: `relation recovery score` $= \frac{B_{i,j}}{\sigma(B)}$, where $\sigma(B)$ denotes the standard deviation of matrix B. Such ratio indicates the extent that the target concept relation entry in the matrix is more significant than a random noise; the larger the score is the more significant the relation recovery.

Results All results are shown in Table 1, which verifies that our proposal can identify the concept-relation of interest from contrastive corpus pairs. For the demonstrated results, we used the same trained model as in the experiments on recovering relations from real-world LLM activations.

A.3.2 Steering Vector Recovery

Except for the relationships between concepts, our model is also able to recover the concepts as current SAEs. To verify this, semi-synthetic benchmarks like SSAE [24] can offer valuable insights into concept identifiability. Following this setting, we tested whether our model can recover steering vectors from paired text. Specifically, we constructed five categories of word pairs where only a single interpretable concept changes, including gender, plurality, comparative, tense, and negation. While these changes are intuitive, ensuring the word pairs capture a clear ground-truth concept is non-trivial. Despite this challenge, our model demonstrated strong performance in identifying the underlying concept differences. Specifically, (1) the average correlation of concept differences within each category exceeded 0.86; (2) assuming one ground-truth pair, the correlation rose above 0.93; and (3) the maximum correlation within a category reached over 0.94. These results support our claim that our model can indeed recover meaningful steering vectors. The word lists for the five categories is summarized in Table 4.

A.4 LLM Activation Experiments

In addition to the experimental results presented in Section 5.3 of the main text, we provide here: (1) detailed settings for training and inference; (2) visualizations of training losses and sparsity values; (3) comparisons across different hyper-parameter settings, and (4) extended experiment on SAEBench with larger latent size and base language models.

A.4.1 Details on the Real-world Experiments Settings

Training We train our linear model on activations from the pretrained LLM `pythia-160m-deduped` [5], using SAELens [6] and dictionary-learning [36] for activation extraction. Importantly, in the original implementation of dictionary-learning [36], activations are loaded using an object named `ActivationBuffer`, which is refreshed with new activations once a predefined consumption threshold is reached. During each refresh, a random shuffling is applied. However, this randomization disrupts the temporal structure of the LLM activations. To preserve temporal information, we modify the corresponding refresh function to disable the random shuffling. Details of this modification can be found in the `examples/README.md` file in our code repository.

The model is trained on a total of 50 million tokens from the `Pile` dataset [17]. To capture time-delayed influences, we consider two values of τ, namely $\{5, 20\}$, as described in Eq. 3. While our main results focus on the setting with $\tau = 20$, which offers better guarantees for capturing rich temporal semantics, this choice will be further justified in a later section of the supplementary materials. To address the distributed and uncertain nature of time-delayed dependencies—where some relations manifest over longer time spans and others over shorter ones—we aggregate the B_τ matrices using max-pooling. This operation preserves any causal link that appears at any time step. We refer to the resulting aggregated matrix as $aggB$. Unless otherwise specified, the weight of the independence constraint on the noise term is set to $\alpha = 0.1$ in Eq. 11.

To better enforce sparsity in the hidden feature activations, we apply TopK filtering [8] in addition to the ℓ_1 sparsity term included in the final loss function. Given the importance of feature dimensionality in Sparse Autoencoders (SAEs), we evaluate three configurations: 768 (which directly matches the LLM's hidden size and aligns with the identifiability discussion in Section 3), 3072, and 6144—the

Table 4: Summary of word pairs in the five categories

Categories	**Pairs**
Gendered Pairs	(male, female), (actor, actress), (prince, princess), (king, queen), (god, goddess), (wizard, witch), (boy, girl), (man, woman), (father, mother), (son, daughter), (brother, sister), (husband, wife), (nephew, niece), (uncle, aunt), (gentleman, lady), (monk, nun), (grandfather, grandmother), (lord, lady), (spokesman, spokeswoman)
Plurality Pairs	(cat, cats), (dog, dogs), (apple, apples), (box, boxes), (child, children), (book, books), (car, cars), (tree, trees), (house, houses), (bird, birds), (chair, chairs), (table, tables), (shoe, shoes), (shirt, shirts), (sock, socks), (cup, cups), (plate, plates), (pen, pens), (bag, bags), (door, doors), (window, windows), (lamp, lamps), (phone, phones), (laptop, laptops), (flower, flowers), (cloud, clouds), (mountain, mountains), (river, rivers), (lake, lakes), (egg, eggs), (grape, grapes), (potato, potatoes), (tomato, tomatoes), (bus, buses), (kiss, kisses), (wish, wishes), (match, matches), (dish, dishes), (baby, babies), (lady, ladies), (city, cities), (party, parties), (family, families), (knife, knives), (leaf, leaves), (wolf, wolves)
Comparative Pairs	(fast, faster), (tall, taller), (small, smaller), (old, older), (young, younger), (short, shorter), (long, longer), (high, higher), (low, lower), (strong, stronger), (weak, weaker), (rich, richer), (poor, poorer), (hard, harder), (soft, softer), (loud, louder), (bright, brighter), (dark, darker), (clean, cleaner), (easy, easier), (happy, happier), (cool, cooler), (deep, deeper), (wide, wider), (narrow, narrower), (thick, thicker), (thin, thinner), (heavy, heavier), (light, lighter), (safe, safer), (cheap, cheaper)
Tense Change Pairs	(walk, walked), (run, ran), (eat, ate), (go, went), (write, wrote), (speak, spoke), (drink, drank), (drive, drove), (read, read), (sleep, slept), (sit, sat), (stand, stood), (fly, flew), (begin, began), (buy, bought), (bring, brought), (build, built), (catch, caught), (choose, chose), (come, came), (cut, cut), (dig, dug), (do, did), (draw, drew), (fall, fell), (feel, felt), (find, found), (get, got), (give, gave), (have, had), (hear, heard), (hide, hid), (hold, held), (keep, kept), (know, knew), (leave, left), (lose, lost), (make, made), (meet, met), (pay, paid), (ride, rode), (say, said), (see, saw), (sell, sold), (send, sent), (sing, sang), (sit, sat), (teach, taught), (think, thought)
Negative Prefix Pairs	(possible, impossible), (legal, illegal), (visible, invisible), (complete, incomplete), (fair, unfair), (known, unknown), (fortunate, unfortunate), (able, unable), (happy, unhappy), (certain, uncertain), (clear, unclear), (real, unreal), (necessary, unnecessary), (likely, unlikely), (available, unavailable), (comfortable, uncomfortable), (pleasant, unpleasant), (reliable, unreliable), (acceptable, unacceptable), (usual, unusual), (wanted, unwanted), (expected, unexpected), (connected, disconnected), (understood, misunderstood), (placed, misplaced)

latter follow the considerations of SAE literature. Note that all of the choices take into account the invertibility condition of the mixing function, as discussed in the footnote of the proof of Theorem 1. We optimize the loss function defined in Eq. 11 using the Adam optimizer with a learning rate of 0.01 and a weight decay of 0.0001. Unless otherwise specified, we use a random seed of 123; additional experiments were conducted with seeds 456 and 789 for robustness.

Inference During inference, our primary goal is to interpret the hidden features—particularly those activated by significant entries in the time-delayed ($aggB$) or instantaneous (M) relation matrices. This selection process differs from conventional SAE interpretation, which typically examines feature importance across the entire feature space by measuring activation strength for a given prompt. In contrast, our method emphasizes the relational structure of features—how they connect to form

semantic transitions. We aim to understand the meaning of each feature by analyzing how both types of relations (instantaneous and time-delayed) link features together.

Our feature selection process involves the following steps: First, we select the top 100 coordinates (we also tried 200, though 100 proved sufficient) from either $aggB$ or M, and extract the corresponding feature dimensions. Next, we generate 10,000 prompts from the `EleutherAI/pile` dataset, convert them into token streams, and feed them into the trained model to observe how each token responds to each selected concept feature. Finally, for each selected feature, we collect the tokens whose activations exceed a threshold (set to 3.0), along with their corresponding prompts. These tokens are viewed as consequences of the activation of the given feature, while the associated prompts serve as contexts that reveal the token and therefore, feature's meaning.

A.4.2 Visualizations of Training Loss and Sparsity Metrics

Here, we compare the training dynamics across different settings by examining the reconstruction loss (Eq. 6), the independence of the estimated noise term (Eq. 9), and the sparsity of both time-delayed and instantaneous relations (Eq. 10). The comparisons are made with respect to variations in hidden feature dimensionality, the sparsity weight on learned relations (i.e., β in Eq.11), the temporal coverage of delayed relations, as determined by $\tau \in \{5, 20\}$, and the parameter of the TopK filtering of the hidden features.

We begin by examining the training dynamics with $\tau = 5$, comparing different settings of the sparsity constraint ($\beta \in \{0.1, 0.01\}$), TopK values ($\{0, 25, 100\}$, where 0 indicates that TopK is disabled), and hidden dimensions (`z_dim` $\in \{768, 3072, 6144\}$). The corresponding results are presented in Figure 8. It is worth noting that certain unstable training batches occasionally impact the overall stability during training. However, since most of the configurations eventually converge and our primary interest lies in the behavior at convergence, we cap the y-axis at 5.0 to improve the clarity of the visualizations. Our key findings are summarized below.

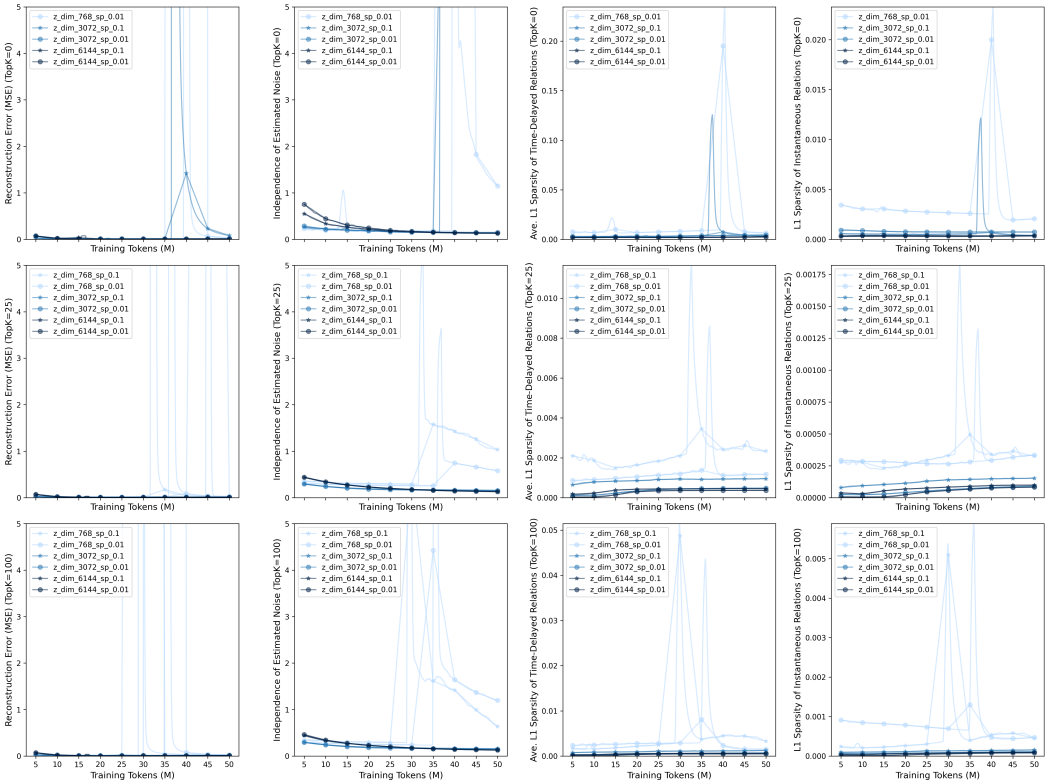


Figure 8: Dynamics of reconstruction loss, noise independence, and time-delayed and instantaneous relations sparsity with setting τ to 5. The x-axis starts at 5M tokens, and the y-axis values are capped at 5 to enhance visualization clarity.

Insights on the Number of Training Tokens and the Impact of Hidden Feature Dimensionality From Figure 8, we observe that 50M training tokens are sufficient for convergence across all settings when the hidden feature dimensionality is greater than 768—specifically, at 3072 and 6144. From the subplots in the first column, it is evident that higher-dimensional hidden features provide greater stability during training. This increased robustness likely helps mitigate the effects of noisy or unstable batches within the token stream, leading to more consistent optimization of the objective. Consequently, in the subsequent case studies, including Section 5.3 of the main content, we focus on the settings with hidden dimensionalities of 3072 and 6144.

Impact of TopK Filtering The training process is in general more stable after applying TopK filtering. More specifically, comparing the sub-diagrams from the first row in Figure 8 to the second and the third rows, we can see that the decrease of the reconstruction error is significantly less effected by some of the token batches, especially, for the setting when feature dimension is set to 3072 or 6144.

Impact of Sparsity Strength In general, when β sets to 0.01 (pay attention to the round marker in Figure 8 as oppose to star marker), both the time-delayed and the instantaneous relations show lower sparsity compared with a stronger sparsity weight. This might be due to a weaker constrain that results a better optimization results, while the stronger one might increase the sharpness of the potential solution space. This also indicates that 0.01 is sufficient for achieving our goal of sparse causal relations in our model.

A.4.3 Sensitivity and Ablation Studies

Sensitivity Study on α and β We conducted additional comparisons with $\beta = 0$, 0.001, 0.005, 0.05, 1.0 and $\alpha = 0$, 0.001, 0.01 to cover a broader hyperparameter range, using $\tau = 5$ and feature dimension 3072. The results are shown in the two tables below, with our selected setting in bold text. The tables highlight that (1) concept relationships are inherently sparse, while a large β disrupts optimization, and (2) α has a stronger effect, with 0.1 being a well-balanced choice.

Table 5: Performance comparison under different values of α

$\boldsymbol{\alpha}$	0.0	0.001	0.01	**0.1**	1.0
Reconstruction Loss ↓	0.0227	0.0191	0.0118	**0.0128**	0.1121
Independence Loss ↓	4.3849	2.4572	0.3910	**0.1448**	0.5252
B_s Sparsity (L1) ↓	0.0012	0.0007	0.0018	**0.0007**	0.0058
M Sparsity (L1) ↓	0.0002	0.0001	0.0002	**0.0001**	0.0009

Table 6: Performance comparison under different values of β

$\boldsymbol{\beta}$	0.0	0.001	0.005	**0.01**	0.05	0.1	1.0
Reconstruction Loss ↓	0.0126	0.0126	0.0126	**0.0128**	0.0126	0.0128	0.8950
Independence Loss ↓	0.1522	0.1496	0.1504	**0.1448**	0.1550	0.1582	3.7682
B_s Sparsity (L1) ↓	0.0003	0.0005	0.0005	**0.0007**	0.0007	0.0013	0.0053
M Sparsity (L1) ↓	0.0001	0.0001	0.0001	**0.0001**	0.0001	0.0002	0.0007

Ablation Studies on Bias Terms Finally, we explore whether there will be potential performance improvement when additional bias terms added to our linear encoder and decoder functions in equation 7, to give a more complete justification of our implementation. We also compared these two settings in the real LLM activations (feature dimension=3072, $\alpha = 0.1$, $\beta = 0.01$, $\tau = 5$). The results shown in the table below indicate that the flexibility gain of the bias terms is not significant in our model.

Table 7: Ablation comparisons on the bias terms for the encoder and decoder

Metric (real-world)	Without Bias	With Bias
Reconstruction Loss	0.0129	0.0129
Independence Loss	0.1452	0.1457
B Sparsity (L1)	0.0007	0.0007
M Sparsity (L1)	0.0001	0.0001

A.4.4 More Showcases on the Recovered Concepts and Relations from LLM Activations

In addition to the examples presented in Section 5.3 of the main text, we provide additional cases here to further illustrate the diversity and interpretability of the recovered concepts and relations, highlighting how they manifest across different domains and contexts.

Table 8: More examples of the discovered time-delayed relations with contextual explanations.

From_ID	From_Explanation	To_ID	To_Explanation	Context
2341	Orders/mandate in appellate judgment	2592	"decision" and "observance"	Legal judgment labels
1856	Technical error message	1833	"FAILURE"	Describes the failure reason
2579	"APPEALS"	2592	Court/party geographical location or case handler	Appeals in legal documents
1833	Ajax request header: 'application', "function" (type, URL, status)	2390	Syntax and functions like "each"	Ajax request function labels
1856	Volume number in case citation	2341	"mandamus" from "writ of mandamus"	Summary of case docket
2100	Page number where case starts	2579	"APPEALS"	Case citation structure
790	Wikipedia ship owner name	2730	"ship"	Wikipedia entity tagging
1825	Email forward/reply dashes	1641	Common words like "subject", "thanks"	Email metadata and signals
1551	Name + "Wynne" (e.g., "John Wynne")	2311	"sat" (in Parliament)	Wikipedia bios for people named Wynne
1124	UTF encoding label	1657	Tags like "UTF-8", "!DOCTYPE"	XML document structure
1675	HTML starting signal "<"	2583	Common HTML tags like "a", "pre"	HTML document recognition
1303	"default" keyword	2623	Follows "default" (e.g., "context", "_")	Generic technical documentation
1895	"Q", "Re", "forward"	1203	"thanks"	Email or Q&A style messages
2708	Personal pronouns ("I", "you")	2584	Tense indicators like "will", "have"	Human language facts

Time-delayed Causal Relations Table 8 showcases further examples of time-delayed causal relations extracted from LLM activations by using our model, with the same setting that we have

Table 9: More examples of the discovered instantaneous relations with contextual explanations.

From_ID	From_Explanation	To_ID	To_Explanation	Context
2341	Labels ‘license’ in comment of "license control prc server"	1856	Labels ‘license’ in both comment and command line	Bash script context
2592	Labels ‘research’	227	Labels ‘research’ with nearby nouns like “program”	Academic texts
2592	Labels ‘magazine’	80	Labels ‘magazine’ and common related nouns like “teenage”, “blogs”	Academic texts
2592	Labels ‘module’	2208	Labels both ‘module’ and ‘exports’ as in “moduleėxports”	JavaScript code
2623	Labels ‘https’	227	Labels both ‘https’ and ‘://’	URLs

shown in the main content Table 3. Many of these reflect the structured nature of legal, technical, and encyclopedic language. For instance, feature 2341 (e.g., “Orders/mandate in appellate judgment”) is linked to feature 2592 (e.g., “decision” and “observance”), revealing how commands or mandates precede judicial conclusions in legal discourse. Similarly, technical logs such as feature 1856 (error messages) anticipate subsequent failure indicators (feature 1833, “FAILURE”), reflecting typical diagnostic progressions in computing contexts.

Notably, semantic connections span heterogeneous domains. Wikipedia entity labeling (e.g., ship names and their categories) and web document structures (e.g., UTF labels leading to encoding declarations) both reveal meaningful temporal dependencies that LLMs internalize. The relation between personal pronouns (feature 2708, “I”, “you”) and tense markers (feature 2584, “will”, “have”) further illustrates how human language patterns are temporally structured, even over several tokens. These cases reinforce the model’s capacity to track and anticipate semantic developments over time in a content- and domain-aware manner.

Instantaneous Causal Relations Table 9 provides more instances of instantaneous relationships, highlighting features that are co-activated within the same context window. In the domain of Bash scripting, we observe co-occurrence between licensing-related comments (feature 2341) and execution commands (feature 1856), showing how LLMs jointly encode comment semantics and imperative script logic.

In academic and technical domains, common conceptual pairs such as “research” and “program”, or “magazine” and related digital terms like “blogs” or “websites”, are represented together (e.g., features 2592 and 227 or 80). These examples suggest that the model forms composite concepts out of frequently co-occurring terms, such as in publication metadata or content descriptions.

In programming contexts, the instantaneous link between “module” (feature 2592) and the JavaScript construct “module.exports” (feature 2208) demonstrates that the model learns the tight coupling between programming keywords. Likewise, the relation between “https” (feature 2623) and its full syntactic pattern “https://” (feature 227) reflects how structured URL formats are stored as unified units in the model’s activation space. Together, these examples demonstrate the model’s ability to encode concise, domain-specific composite structures through simultaneous feature activation.

Notes on the Results Following our presentation of the causal relations recovered from LLM activations, we clarify several key points regarding the interpretation of these results. First, due to variations in tokenization strategies across different corpora, many identified tokens in a given sentence may correspond only to partial words. This issue can be exacerbated by noise introduced during data collection processes such as OCR or web crawling. To address this, we rely on human judgment and linguistic intuition to infer and annotate the complete underlying word, ensuring that

Table 10: Gemma-2-2B instantaneous-relation-only model on SAEBench with different latent sizes.

Latent Size	Recon. Loss ↓	Sparse Probing Acc. ↑	Absorption ↓	Autointerp ↑
6k	0.0108	0.6736	0.0139	0.6883
16k	0.0059	0.6918	0.0167	0.7117

Table 11: Absorption statistics with extended training budgets.

Model	Full Absorption Fraction	Absorption Fraction	# Split Features
Pythia-160M-16k	6.471×10^{-2}	9.185×10^{-3}	1.043
Gemma-2-2B-16k	1.289×10^{-2}	3.794×10^{-4}	1.269

the labeling remains accurate and avoids overextending to unrelated tokens. Second, the recovered time-delayed relations we present may be somewhat semantically constrained, as the clearest relations tend to align with explicit syntactic structures. Many of our examples—such as those from code snippets or legal documents—convey semantic information through formal syntax. While these cases are illustrative, we view the discovery of more abstract, syntactically diffuse relations in general language text as an important direction for future work. It is also important to note that the examples we present were not cherry-picked; rather, they are representative cases that naturally appear throughout the dataset and were surfaced by our method. These relational patterns would not be easily discoverable using sparse autoencoders (SAEs), as SAEs do not consider interactions between features. Finally, we observe that feature pairs exhibiting strong causal relations tend to be activated under highly similar prompt conditions, indicating that these features are contextually aligned and often co-occur within the same linguistic environments.

A.4.5 Additional SAEBench Results on Larger Latent Sizes and Models

Following the same dataset and training protocol as in the main experiments, we trained the simplified instantaneous-relation-only variant with 16k latents on Gemma-2-2B and compared it to our 6k-latent configuration. As shown in Table 10, the 16k model reduces reconstruction loss (0.0059 vs. 0.0108) and slightly improves sparse probing top-1 accuracy (0.6918 vs. 0.6736). Its absorption score is modestly higher (0.0167 vs. 0.0139) but remains small, and the Autointerp score increases (0.7117 vs. 0.6883). Overall, performance on SAEBench metrics remains at a similar level across latent sizes.

We also extended the training scale to 500M tokens for Pythia-160M and to 300M tokens for Gemma-2-2B, both with 16k latents. In both cases we observed very small absorption fractions, and the mean number of split features remained close to one, indicating minimal feature splitting. Summary statistics are reported in Table 11.

A.4.6 Statistical Testing and Absorption Analysis

To strengthen the empirical findings, we additionally performed statistical testing to assess the equivalence of reconstruction losses and the robustness of absorption scores. Using 100 samples per method ($N = 300$), any shift ≥ 0.00127 across groups would be detected with power ≥ 0.8. Pairwise Welch–TOST and Hodges–Lehmann tests with $\Delta = 0.001$ confirmed equivalence: all 90% confidence intervals lay within $[-\Delta, \Delta]$, demonstrating statistical equivalence at $\alpha = 0.05$ among the three methods.

For absorption, although rigorous hypothesis testing is challenging due to the very small magnitudes observed, we collected a sufficiently large number of samples (≥ 200) to establish confidence intervals. The mean and 95% confidence intervals were 0.0135 ± 0.0002 for the 6k model and 0.0136 ± 0.0002 for the 16k model, which are more than sufficient to demonstrate negligible absorption in practice.

Additionally the signal-to-noise ratio (20.02 for our method vs. 2.39 for the SAE baseline) already indicates a strong margin. Such a large difference is unlikely to arise from random noise.

A.4.7 Preliminary Investigation with Time Lag up to 100

To address the potential limitation that a fixed value of τ may be overly restrictive in capturing the rich and diverse semantics of real-world contexts, we explore a more flexible approach. Specifically, different types of concept-relations may require varying numbers of steps to be successfully recovered. Furthermore, even for a single concept-relation, stable recovery across different contexts may necessitate a range of steps rather than a single fixed value. In light of these considerations, in addition to the recovered relations shown in Table 3, Table 8, and Table 9, we present the relations captured within 100 steps, grouping them into bins of sizes 10, 20, and 50. This binning naturally categorizes the relations of interest, facilitating further analysis and discussion.

The increased flexibility provided by a larger time lag allows us to recover a greater number of concept-relations. For example, we can recover relations such as "monument" $\rightarrow$ "from" and "seek" $\rightarrow$ "opportunity". Interestingly, increasing the time lag not only allows longer-range relations to be captured but also enables previously overlooked relations to be discovered, as this flexibility improves identification of concepts entangled in the relation. To better illustrate the relations recovered with a larger time lag, we are preparing a web demo, which will soon be included in the code repository once it is ready.

To better illustrate the relations recovered with a larger time lag, we are preparing a web demo, which will soon be included in the code repository once it is ready. However, our primary contribution is to demonstrate that our model can recover relation-concepts more effectively than existing SAEs, addressing a gap that is currently missing but crucial for advancing LLM interpretability. A broader and more systematic study of this phenomenon is left to future work.

A.4.8 Addition Experiments with Pretrained SAE

As ablation study we additionally construct our linear model using the pretrained Sparse Autoencoder (SAE) from Gemma Scope [29] on the Gemma 2 2B model [53]. To enable feasible qualitative evaluation, we selected the top $2,034$ most frequently activated features from the commonly used SAE `gemma-2-2b/20-gemmascope-res-16k` using the SAELens package [6]. We trained our linear model on 5 million tokens from the `Pile` [17] dataset.

Since time-delayed influences may occur with variable time lags, we set a sufficiently large value for τ in Eq. 3. In practice, we use $\tau \leq 20$ and aggregate the time-delayed matrices B_τ using max-pooling—that is, if a causal link exists in any of the time-lagged matrices B_τ, we consider that link to be present in the aggregated causal structure.

Case Studies Our analysis reveals rich causal structures among programming-related concepts in LLM activations. We examine both time-delayed and instantaneous causal relationships, providing insights into how the model processes and generates code-related content.

Time-Delayed Causal Relations We identified several meaningful time-delayed causal relationships in programming contexts. A prominent example is the causal link from a concept representing "function definitions and related code structure in programming languages" to a concept representing "variable definitions and data types in programming contexts." This relationship aligns with the natural structure of programming, where global function definitions often precede and influence local variable declarations or data structures. When the model processes or generates function definitions, it subsequently activates concepts related to the variables and data types that would appear within those functions.

Additional time-delayed relationships include causal links from "programming language syntax specifications" to "code implementation details" and from "algorithmic problem statements" to "solution implementation structures." These relationships demonstrate how the model captures the sequential dependencies inherent in programming tasks, where understanding of requirements or specifications precedes implementation details.

Instantaneous Causal Relations Our method also reveals interesting instantaneous causal relationships that occur within the same time step. We observe a strong instantaneous causal link between a concept representing "specific formatting and notation elements commonly used in mathematical expressions or programming syntax" and a concept representing "mathematical symbols

and expressions in technical content." This relationship indicates that the model simultaneously processes formatting rules and the mathematical content they structure, reflecting how these aspects are intrinsically connected in code representation.

We also identified instantaneous causal relationships between "programming language keywords" and "syntax highlighting patterns," as well as between "code indentation patterns" and "block structure delineation." These instantaneous relationships capture the syntactic constraints that operate simultaneously within programming languages, where certain elements must co-occur for the code to be well-formed.

These case studies demonstrate that our method can extract meaningful causal relationships from real LLM activations, providing insights into how these models process and generate structured content like code. The identified causal structures align with the logical and syntactic relationships one would expect in programming contexts, validating the effectiveness of our approach for interpretability research.

A.5 Compute Resources and Code

All experiments were conducted on a computing cluster equipped with NVIDIA L40 GPUs. The synthetic verification experiments were run using 16 CPU cores, 32 GB of memory, and a single GPU. The Jacobian complexity experiment was executed on CPU only, as the computation did not fit within GPU VRAM; to avoid out-of-memory (OOM) errors, 32 CPU cores and 400 GB of memory were allocated. The scaled-up synthetic experiment with the linear model used 32 CPU cores, 64 GB of memory, and one GPU. The large language model (LLM) activation experiment was performed using 16 CPU cores, 15 GB of memory, and a single GPU.

The code that can replicate the main experiments presented in our paper can be accessed via `https://github.com/xiangchensong/temp-inst-sae`

A.6 Limitations

We acknowledge certain limitations of our work. The linear approximation, while computationally efficient and theoretically grounded, may not capture all nonlinear interactions present in LLM activations. Future work should explore extending our framework to incorporate bounded nonlinearities while maintaining computational tractability. Additionally, developing methods to automatically interpret the discovered causal structures in terms of human-understandable concepts remains a challenge. Our method also assumes a specific form of temporal dependency that might not fully capture the long-range dependencies that LLMs can handle. The current formulation is limited to first-order temporal dependencies, and extending this to higher-order dependencies would increase computational complexity. Lastly, tokenization has been shown to critically affect LLM identifiability during our evaluations, even though it is not inherently part of LLM interpretation methods. We emphasize the importance of choosing a tokenization strategy that preserves semantic information and maximizes the effectiveness of LLM interpretation approaches.

A.7 Societal Impacts

Our interpretability approach can improve transparency, support alignment interventions, facilitate debugging and bias detection, advance scientific understanding of causal representations, and inform educational tools that raise AI literacy. At the same time, deeper insight into model internals may enable malicious manipulation, create misplaced confidence in safety tools, widen resource disparities, expose private information from training data, and distract attention from broader social and governance measures. Future work should include collaboration with ethicists, social scientists, and policy experts to guide responsible use.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Major claims are described in abstract and emphasized in introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation is discussed in Appendix A.6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theorems are supported with complete and correct proof in Appendix A.1 with assumptions clearly presented in main text.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: A detailed description of the experimental setup is provided in Appendix A.2 for the synthetic experiments and in Appendix A.4 for the LLM activation experiments. The codebase required to reproduce the experiments is included in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The model data used in our experiments are either from publicly available datasets or can be generated using the codebase provided in the supplementary materials. The main experimental results can be reproduced using this submitted codebase.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed settings are provided in the Appendix A.2 and A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The mean value of multiple runs and with std plotted with error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes the computation resources use in the experiment is provided in Appendix A.5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: authors have reviewed and conform the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in a seperate section in Appendix A.7

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets including baseline codes and the dataset and models are explicitly mentioned and credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Detailed instructions have been provided along with the codebase.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA] .

    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.