

A Technical Appendices and Supplementary Material

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} 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 diagonal 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 $T = PD$.*

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

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

Set $D = \text{diag}(\alpha_1, \dots, \alpha_n)$.

For every basis vector e_i we have

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

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 $D_1 = D_2$. \square

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

$$T = PD,$$

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

We also include the discussion for higher order generalization which is originally given by [26] as follows: For any given τ , and subsequence which is centered at \mathbf{z}_t with previous l_o and following h_i steps, i.e., $\mathbf{c}_t = \{\mathbf{z}_{t-l_o}, \dots, \mathbf{z}_t, \dots, \mathbf{z}_{t+h_i}\} \in \mathbb{R}^{(l_o+h_i+1) \times n}$. In this case, the vector function $w(i, j, m)$ in Sufficient Variability Assumption should be modified as

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

⁴Note that for the case when the dimension of \mathbf{x} matches the dimension of \mathbf{z} , then the bijection assumption in [26] 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, 49].

Besides, $2 \times n \times (lo + hi + 1) + |\mathcal{M}_{c_t}|$ values of linearly independent vector functions in $z_{t',m}$ for $t' \in [t - lo - 1, \dots, t - lo - \tau]$ and $m \in [1, \dots, n]$ 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:

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

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:

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

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

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:

$$M = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0.5 & 0 & 0 & \dots & 0 \\ 0 & 0.5 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0.5 & 0 \end{bmatrix} \quad (16)$$

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 LLM Activation Experiments

In addition to the experimental results presented in Section 5.2 of the main text, we provide here: (1) detailed settings for training and inference; (2) visualizations of training losses and sparsity values; and (3) comparisons across different hyper-parameter settings.

A.3.1 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 [34] for activation extraction. Importantly, in the original implementation of dictionary-learning [34], 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 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 3. 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.3.2 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.

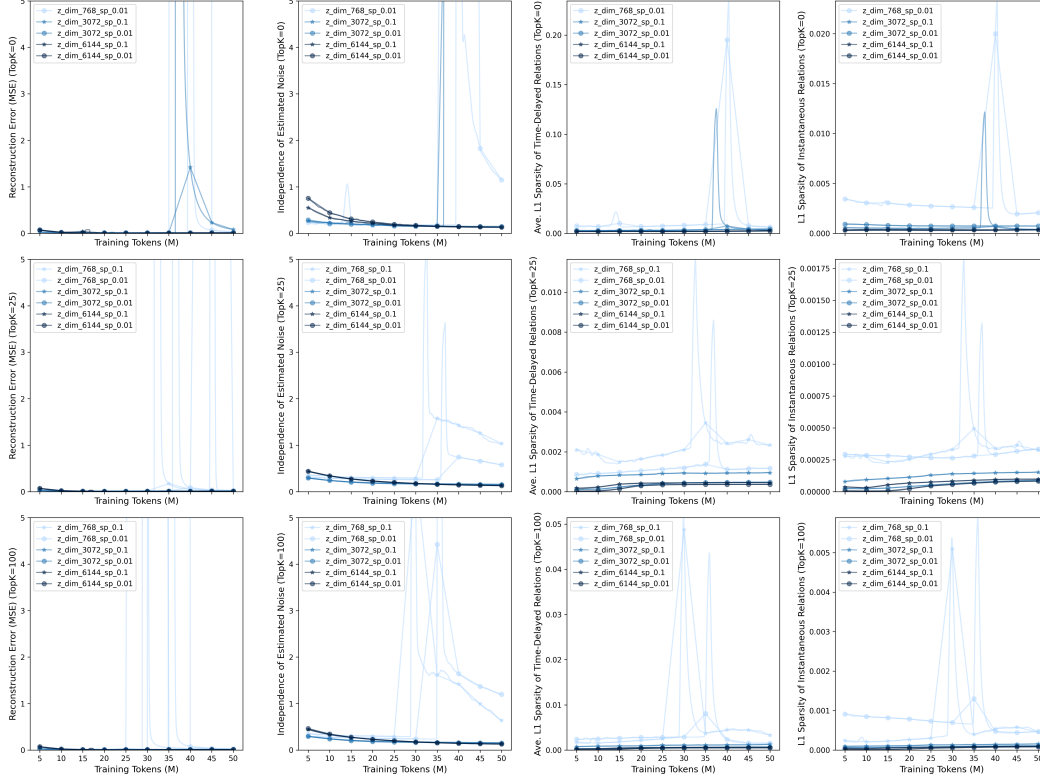


Figure 8: Dynamics of reconstruction loss, noise independence, and time-delayed and instantaneous relations sparsity with setting τ to 5. The x-axis starts at 5M tokens, and the y-axis values are capped at 5 to enhance visualization clarity.

804 Insights on the Number of Training Tokens and the Impact of Hidden Feature Dimensionality

805 From Figure 8, we observe that 50M training tokens are sufficient for convergence across all settings
 806 when the hidden feature dimensionality is greater than 768—specifically, at 3072 and 6144. From
 807 the subplots in the first column, it is evident that higher-dimensional hidden features provide greater
 808 stability during training. This increased robustness likely helps mitigate the effects of noisy or
 809 unstable batches within the token stream, leading to more consistent optimization of the objective.
 810 Consequently, in the subsequent case studies, including Section 5.2 of the main content, we focus on
 811 the settings with hidden dimensionalities of 3072 and 6144.

812 **Impact of TopK Filtering** The training process is in general more stable after applying TopK
 813 filtering. More specifically, comparing the sub-diagrams from the first row in Figure 8 to the second
 814 and the third rows, we can see that the decrease of the reconstruction error is significantly less effected
 815 by some of the token batches, especially, for the setting when feature dimension is set to 3072 or
 816 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.

Impact of the Choice of τ We further explore the impact of the choice of τ , which represents a trade-off: a larger τ allows the model to capture a broader range of time-delayed dependencies among hidden features, while a smaller τ simplifies the optimization by reducing the number of parameters involved. As a preliminary step to examine this trade-off, we present the training dynamics for the setting with $\tau = 20$ in Figure 9 as a direct comparison to Figure 8, where τ is fixed at 5.

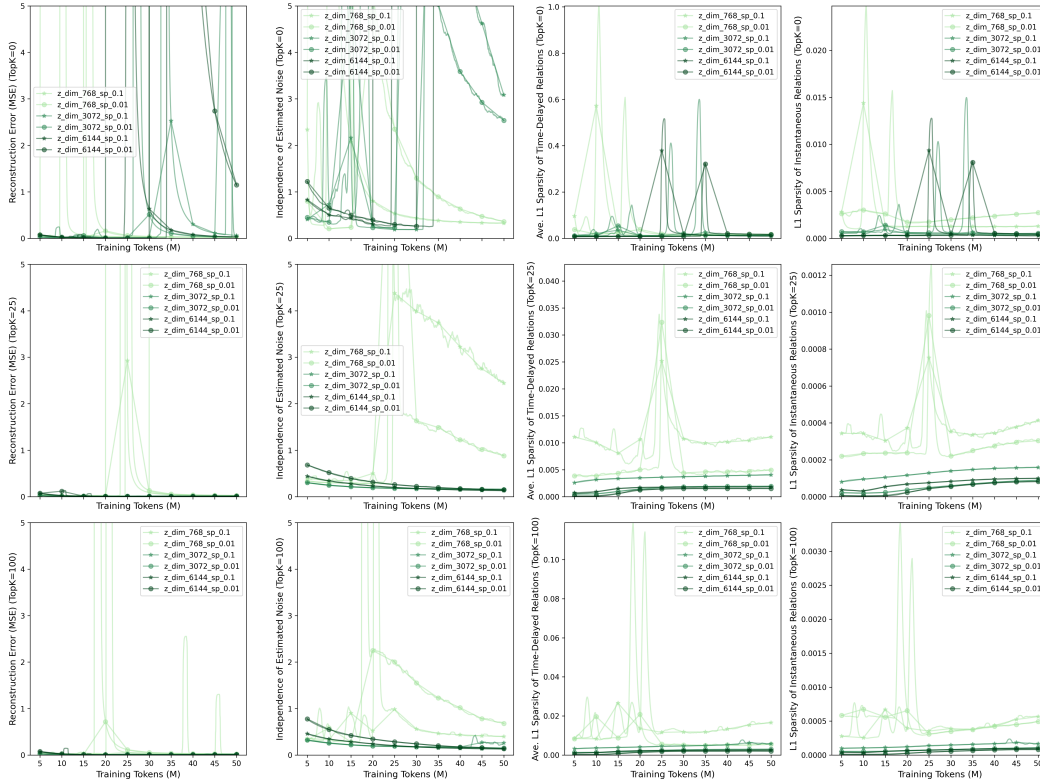


Figure 9: Dynamics of reconstruction loss, noise independence, and time-delayed and instantaneous relations sparsity with setting τ to 20. The x-axis starts at 5M tokens, and the y-axis values are capped at 5 to enhance visualization clarity and ensure a fair comparison across settings.

In comparison, we observe a subtle yet consistent difference: models with a larger τ tend to encounter unstable training batches earlier than those with a smaller τ , likely due to their broader temporal window for capturing dependencies among hidden features. Nevertheless, both configurations eventually converge. Notably, a larger τ often leads to earlier convergence, as it enables the model to address instability sooner in the training process.

It is also worth noting that, considering the typical sentence lengths in standard training corpora, our aim to capture rich semantic relationships among hidden features, and the manageable computational cost achieved through our linearized setup, we chose $\tau = 20$ for our experiments.

A.3.3 More Showcases on the Recovered Concepts and Relations from LLM Activations

In addition to the examples presented in Section 5.2 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 2: 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	“APPEALS”	Appeal labels in legal documents
2592	Court region or case handler	1833	Ajax request function/meta header	Geographical or technical request context
1833	Ajax application function header	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

Table 3: 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 “module.exports”	JavaScript code
2623	Labels ‘https’	227	Labels both ‘https’ and ‘:/'	URLs

Time-delayed Causal Relations Table 2 showcases further examples of time-delayed causal relations extracted from LLM activations by using our model, with the same setting that we have shown in the main content Table 1. 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 3 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 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.

888 A.3.4 Addition Experiments with Pretrained SAE

889 As ablation study we additionally construct our linear model using the pretrained Sparse Autoencoder (SAE) from Gemma Scope [27] on the Gemma 2 2B model [51]. 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 SAE lens package [6]. We trained our linear model on 5 million tokens from the Pile [17] dataset.

894 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.

898 **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.

901 **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.

909 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.

914 **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.

921 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.

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

931 **A.4 Compute Resources and Code**

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

939 The code that can replicate the main experiments presented in our paper can be ac-
940 cessed in the supplementary material folder and via [https://anonymous.4open.science/r/](https://anonymous.4open.science/r/temp-inst-sae-neurips-2025-C465)
941 [temp-inst-sae-neurips-2025-C465](https://anonymous.4open.science/r/temp-inst-sae-neurips-2025-C465).

942 **A.5 Limitations**

943 We acknowledge certain limitations of our work. The linear approximation, while computationally
944 efficient and theoretically grounded, may not capture all nonlinear interactions present in LLM acti-
945 vations. Future work should explore extending our framework to incorporate bounded nonlinearities
946 while maintaining computational tractability. Additionally, developing methods to automatically
947 interpret the discovered causal structures in terms of human-understandable concepts remains a
948 challenge. Our method also assumes a specific form of temporal dependency that might not fully
949 capture the long-range dependencies that LLMs can handle. The current formulation is limited to
950 first-order temporal dependencies, and extending this to higher-order dependencies would increase
951 computational complexity. Furthermore, our evaluation on real LLM data provides qualitative insights
952 but would benefit from more rigorous quantitative evaluation metrics.

953 **A.6 Societal Impacts**

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

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] 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.
- [25] 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.
- [26] 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.
- [27] Tom Lieberum, Senthoran 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.
- [28] Tom Lieberum, Senthoran 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.
- [29] 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.
- [30] 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.
- [31] 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.

- [32] Aleksandar Makelov, George Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control, 2024.
- [33] Luke Marks, Alasdair Paren, David Krueger, and Fazl Barez. Enhancing neural network interpretability with feature-aligned sparse autoencoders, 2024.
- [34] Samuel Marks, Adam Karvonen, and Aaron Mueller. dictionary_learning. https://github.com/saprmarks/dictionary_learning, 2024.
- [35] 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.
- [36] 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.
- [37] Hiroshi Morioka and Aapo Hyvärinen. Causal representation learning made identifiable by grouping of observational variables. *arXiv preprint arXiv:2310.15709*, 2023.
- [38] 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.
- [39] 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.
- [40] Neel Nanda. Open Source Replication & Commentary on Anthropic’s Dictionary Learning Paper, Oct 2023.
- [41] 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.
- [42] 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.
- [43] Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models, 2024.
- [44] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.
- [45] 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.
- [46] 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.
- [47] 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.
- [48] 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.
- [49] Yuchen Sun and Kejun Huang. Global identifiability of overcomplete dictionary learning via l1 and volume minimization. In *The Thirteenth International Conference on Learning Representations*.

- 1101 [50] Glen M. Taggart. Prolu: A nonlinearity for sparse autoencoders,
 1102 2024. [https://www.alignmentforum.org/posts/HEpufTdakGTTKgoYF/
 1103 prolu-a-nonlinearity-for-sparse-autoencoders](https://www.alignmentforum.org/posts/HEpufTdakGTTKgoYF/prolu-a-nonlinearity-for-sparse-autoencoders)
- 1104 [51] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin,
 1105 Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé,
 1106 et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint*
 1107 *arXiv:2408.00118*, 2024.
- 1108 [52] Andrew Templeton, Timothy Conerly, Jacob Marcus, John Lindsey, Tamera Bricken, Bowen
 1109 Chen, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet.
 1110 Technical report, Anthropic, 2024. Transformer Circuits Thread Technical Report.
- 1111 [53] Constantin Venhoff, Anisoara Calinescu, Philip Torr, and Christian Schroeder de Witt. Sage:
 1112 Scalable ground truth evaluations for large sparse autoencoders, 2024.
- 1113 [54] Julius von Kügelgen. Identifiable causal representation learning: Unsupervised, multi-view, and
 1114 multi-environment. *arXiv preprint arXiv:2406.13371*, 2024.
- 1115 [55] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt.
 1116 Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.
- 1117 [56] Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning.
 1118 *arXiv preprint arXiv:2210.13647*, 2022.
- 1119 [57] Weiran Yao, Yuewen Sun, Alex Ho, Changyin Sun, and Kun Zhang. Learning temporally causal
 1120 latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*, 2021.
- 1121 [58] Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from
 1122 multiple distributions: a general setting. In *Proceedings of the 41st International Conference on*
 1123 *Machine Learning*, ICML’24. JMLR.org, 2024.