TEMPORAL SPARSE AUTOENCODERS: LEVERAGING THE SEQUENTIAL NATURE OF LANGUAGE FOR INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Translating the internal representations and computations of models into concepts that humans can understand is a key goal of interpretability. While recent dictionary learning methods such as Sparse Autoencoders (SAEs) provide a promising route to discover human-interpretable features, they often only recover token-specific, noisy, or highly local concepts. We argue that this limitation stems from neglecting the temporal structure of language, where semantic content typically evolves smoothly over sequences. Building on this insight, we introduce Temporal Sparse Autoencoders (T-SAEs), which incorporate a novel contrastive loss encouraging consistent activations of high-level features over adjacent tokens. This simple yet powerful modification enables SAEs to disentangle semantic from syntactic features in a self-supervised manner. Across multiple datasets and models, T-SAEs recover smoother, more coherent semantic concepts without sacrificing reconstruction quality. Strikingly, they exhibit clear semantic structure despite being trained without explicit semantic signal, offering a new pathway for unsupervised interpretability in language models.

1 INTRODUCTION

Interpretability aims to translate the internal representations and computations of language models into concepts that humans can understand, evaluate, and ultimately control. In practice, the most useful insights often involve high-level drivers of model behavior, such as the semantics a model encodes or the state it is operating in, rather than surface-level statistical patterns. Recent dictionary learning methods, such as Sparse Autoencoders (SAEs), have shown promise in explaining language models. By projecting dense latent representations into a sparse, human-interpretable feature space, SAEs enable both the recovery of known linguistic patterns and the discovery of novel concepts within models.

However, when applied to large language models (LLMs), SAEs frequently fall short of this goal. The features they recover are often token-specific, local, and noisy, capturing superficial syntactic patterns (e.g., "`the phrase 'The' at the start of sentences`"[1] or "`Sentence endings or periods`," Figure 1) rather than coherent, high-level semantic concepts. One interpretation of this phenomenon is that LLMs themselves fail to encode deep semantic structure, functioning instead as sophisticated next-token predictors. A more plausible explanation, however, is that current concept discovery methods are inadequate; their design biases them toward recovering shallow patterns even when richer structure exists in the underlying representations. We argue that this limitation stems from a fundamental issue with how SAEs are formulated. Human language is inherently structured: meaning is conveyed through context and semantics that evolve smoothly over time, while syntax is governed by more local dependencies. Yet current dictionary learning methods ignore this sequential structure, treating tokens as independent and stripped of context.

To address this gap, we introduce the notion of temporal consistency, or the property that high-level semantic features of a sequence remain stable over adjacent (or nearby) tokens, while low-level syntactic features may fluctuate more rapidly. For example, in the sentence *"Photosynthesis is the*

[1] Neuronpedia, Feature 11795 of Gemmascope's Gemma2-2b, Residual, 16k SAE

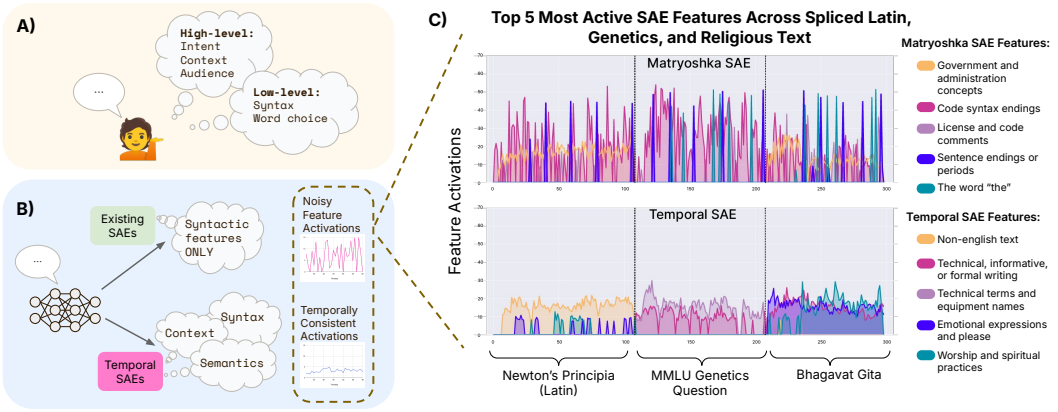

Figure 1: **A)** Human language production involves high-level features such as semantic content and surrounding context, as well as low-level features such as syntactical requirements and specific word choices. **B)** While existing SAEs mostly recover syntactic information, Temporal SAEs balance recovery of semantics, syntax, and context. **C)** When decomposing a sequence composed of three passages: Newton's Principia, an MMLU genetics question, and the Bhagavat Gita, Temporal SAEs (bottom) are able to smoothly detect the semantic shifts in the passage, with highly active features strongly correlating to the true content of the text, whereas existing SAEs (such as Matryoshka, top), are much noisier, varying on almost a per-token basis, and do not easily depict these shifts.

process by which plants convert sunlight into energy," a temporally consistent semantic feature might represent "discussion of plant biology" or "scientific explanation." This feature should remain active throughout the entire sentence, because the semantic content does not change from word to word. In contrast, syntactic features such as "capitalized first word" or "plural noun" activate only at specific tokens (e.g., "Photosynthesis," "plants"), reflecting local rather than global structure.

Building on this principle, we propose Temporal Sparse Autoencoders (T-SAEs), a simple modification to SAEs that incorporates a temporal contrastive loss encouraging high-level features to activate consistently over adjacent tokens (Figure 1). This modification, grounded in linguistic intuition, enables SAEs to more reliably capture semantic features while still disentangling them from low-level syntactic ones. Despite being trained only on a self-supervised context-similarity objective, T-SAEs yield representations with significantly improved semantic structure, without requiring any explicit semantic signal. Through experiments, we show that T-SAEs consistently recover higher-level semantic and contextual concepts, exhibit smoother and more temporally consistent activations over sequences, and remain competitive with existing SAEs on standard benchmarks such as reconstruction quality.

Our contributions are the following:

1. We introduce a simple data-generating process for language that distinguishes between high-level, temporally consistent semantic variables and low-level, local syntactic variables. This framework formalizes how we expect language models to encode linguistic information and provides guidance for designing better interpretability methods.
2. Building on this framework, we propose Temporal SAEs, which partition latent features into semantic and syntactic components. We introduce a novel temporal contrastive loss which enforces consistency of high-level activations across sequences, encouraging T-SAEs to disentangle semantic and syntactic features in a self-supervised manner.
3. Through experiments on multiple models and datasets, we show that T-SAEs: a) Recover semantic and contextual information more reliably than existing SAEs, b) Exhibit improved disentanglement between high- and low-level features, c) Maintain competitive performance on standard reconstruction benchmarks, and d) Provide practical interpretability benefits, including a case study on safety-related concepts.

We release our code, trained T-SAEs, and interpreted latents for reproducibility[2].

[2]Link will be provided upon acceptance.

2 Related Work

Sparse Autoencoders. In recent years, SAEs have emerged as a popular mechanistic interpretability technique for self-supervised concept discovery (Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024). While they were initially promising for addressing the problem of polysemanticity, where a single neuron in a model can represent multiple concepts (Elhage et al., 2022), in practice, they have been shown to create new problems, such as feature splitting and absorption (Chanin et al., 2024), where concepts are split across multiple features or absorbed into less interpretable sub-features. To address these subsequent issues, methods such as Matryoshka SAEs (Bussmann et al., 2025) and Transcoders (Paulo et al., 2025) have been proposed, which learn hierarchical and causal features. Recent work has also proposed learning dictionary features that are constrained to the data manifold (Fel et al., 2025) and reflect intuition about the geometry of model latent spaces (Hindupur et al., 2025), allowing for the recovery of heterogeneous concepts. However, all of these works assume a fully unsupervised objective for learning SAEs, treating each token in the training data as i.i.d., without acknowledging the temporal aspect of language and other sequential modalities. As a result, SAEs are known to suffer from a variety of problems, including "dense" activation behavior (Sun et al., 2025) and lack of utility for steering (Bhalla et al., 2025; Wu et al., 2025).

Linguistics, Cognitive Science, and Neuroscience. Many foundational works in linguistics study the relationship between syntactic structure and semantic meaning (Chomsky, 1965), with nearly all major theories recognizing a fundamental distinction between semantics and syntax. Evidence of the two having distinct representations has been found in developmental psychology (Brown, 1973) and neuroscience (Neville et al., 1992; Zhang et al., 2025). In computational linguistics, separate approaches emerged to model language purely syntactically via Hidden Markov Models (Manning & Schutze, 1999) or through a bag-of-words approach to model topics with Latent Dirichlet Allocation (Blei et al., 2003). Griffiths et al. (2004) combine the two into a single model consisting of an HMM where one "semantic" class denotes a topic model which samples words in an LDA-like fashion. Importantly, Griffiths et al. (2004) argue that semantics in language exhibit long-range behavior, with different words or sentences in the same document having similar semantic content, whereas syntax is mostly dependent on short-range interactions, motivating our method.

Dictionary learning with natural priors. Before dictionary learning was formalized, (Olshausen & Field, 1996) showed that decomposing natural images in a linear sparse manner leads to Gabor-like receptive filters, similar to what is found in the visual cortex without any priors on visual data. More recent approaches realized the benefit of taking data priors into the dictionary learning process. For instance, low-rankness has proved an effective parsimonious prior beyond sparsity (Davenport & Romberg, 2016; Vu & Monga, 2017), multi-scale structure in medical imaging (Ong & Lustig, 2016), and Luo et al. (2019) argued that sequential frames in videos exhibit temporal consistency. Our work draws parallels to this trajectory of research in compressive sensing and dictionary learning by introducing structural priors to the unsupervised learning of SAEs.

3 Our Framework: Temporal Sparse Autoencoders

3.1 Formulating the Data Generating Process

Consider a speaker who is producing language, or a sequence of tokens $\tau_1, ..., \tau_T$. When the speaker produces each token τ_t, they take into account many factors — their intent in speaking, the prior context of the token (i.e., what has already been said), syntactic requirements, and other implicit features corresponding to speaker idiosyncrasies (such as their accent, their method of language production, or linguistic style). These factors can be modeled as latent variables that control the language generation process, and they can be generally categorized into two types: variables that encode *high-level* or *global* information, $\mathbf{h}_t$, and variables that encode *low-level* or *local* information $\mathbf{l}_t$. High-level variables can be thought of as features that are invariant to the specific token, such as those capturing semantics and intent. Conversely, low-level information pertains to the specific timestep or token being produced, such as a word's grammatical gender.

We model the speaker's language production process as a function mapping the context and these latent variables to the next token

$$\tau_t = \phi(\tau^{t-1}, \mathbf{h}_t, \mathbf{l}_t),$$

where τ^{t-1} represents the previously-uttered tokens $\tau_1, ..., \tau_{t-1}$. We pass tokens τ^T into a language model which produces latent vectors $\{\mathbf{x}_t^L\}_{t=1}^T \in \mathbb{R}^d$ at layer L. For simplicity, we analyze a single layer and drop the L superscript. We assume that the model represents $\mathbf{h}_t$ and $\mathbf{l}_t$ through an invertible mapping g such that $g(\mathbf{h}_t, \mathbf{l}_t) = \mathbf{x}_t$. Our goal is to recover the encoding of the data-generating latent variables by decomposing its representations into interpretable features corresponding to $\mathbf{h}_t, \mathbf{l}_t$. To do so, we make the following key assumptions:

Assumption 1 (Temporal Consistency.)**.** *$\mathbf{h}_t$ is time invariant, meaning two tokens $\mathbf{x}_t, \mathbf{x}_{t'}$ sampled from the same sequence should have similar latents $\mathbf{h}_t \approx \mathbf{h}_{t'}$.*

Assumption 2 (Hierarchical Representation of Features.)**.** *The mapping g is hierarchical in the sense that it can operate on just $\mathbf{h}_t$ and satisfies $0 = \|g(\mathbf{h}_t, \mathbf{l}_t) - \mathbf{x}_t\| \leq \|g(\mathbf{h}_t) - \mathbf{x}_t\| \leq \epsilon$. In other words, $\mathbf{h}_t$ can reconstruct $\mathbf{x}_t$, but $\mathbf{l}_t$ contains signal about $\mathbf{x}_t$ that is not explained by $\mathbf{h}_t$.*

3.2 Temporal Sparse Autoencoders

We partition the SAE feature space into high-level and low-level features. Without loss of generality, we assume the first h indices are our high-level features and the last $m - h$ indices are our low-level features, where m is the number of features in the SAE. The SAE architecture can be defined as the following, taking in input $\mathbf{x}_t \in \mathbb{R}^d$:

$$\mathbf{f}(\mathbf{x}_t) = \sigma(\mathbf{W}^{\text{enc}}\mathbf{x}_t + \mathbf{b}^{\text{enc}}),$$
$$\hat{x}(\mathbf{f}) = \mathbf{W}^{\text{dec}}\mathbf{f}(\mathbf{x}_t) + \mathbf{b}^{\text{dec}}.$$

Here, $\mathbf{W}^{\text{enc}} \in \mathbb{R}^{m\times d}$ is the encoder matrix, and $\mathbf{W}^{\text{dec}} \in \mathbb{R}^{d\times m}$ is the decoder comprised of high-level features $\mathbf{W}^{\text{dec}}_{0:h} \in \mathbb{R}^{d\times h}$ and low-level features $\mathbf{W}^{\text{dec}}_{h:m} \in \mathbb{R}^{d\times(m-h)}$ such that their concatenation equals $\mathbf{W}^{\text{dec}}$. The encoder and decoder bias are $\mathbf{b}^{\text{enc}} \in \mathbb{R}^d$ and $\mathbf{b}^{\text{dec}} \in \mathbb{R}^d$, respectively. We define the following loss function, where the high-level features $\mathbf{f}_{0:h}(\mathbf{x}_t)$ should reconstruct the input and the low-level features $\mathbf{f}_{h:m}(\mathbf{x}_t)$ should reconstruct the residual, as discussed in Assumption 2, similar to the Matryoshka SAE objective in (Bussmann et al., 2025).

$$\mathcal{L}_{\text{matr}}(\mathbf{x}_t) = \mathcal{L}_H + \mathcal{L}_L,$$
$$\mathcal{L}_H = \|\mathbf{x}_t - \mathbf{W}^{\text{dec}}_{0:h}\mathbf{f}_{0:h}(\mathbf{x}_t) + \mathbf{b}^{\text{dec}}\|_2^2,$$
$$\mathcal{L}_L = \|\mathbf{x}_t - \mathbf{W}^{\text{dec}}\mathbf{f}(\mathbf{x}_t) + \mathbf{b}^{\text{dec}}\|_2^2.$$

We then add a training objective that encourages $\mathbf{W}^{\text{enc}}_{0:h}$ to learn temporally-consistent features following Assumption 1 about $\mathbf{h}_t$: high-level features should be similar for two tokens from the same sequence, particularly for two adjacent tokens. To enforce this, we add a contrastive term to the loss function that encourages $\mathbf{W}^{\text{enc}}_{0:h}\mathbf{x}_t$ to be similar to $\mathbf{W}^{\text{enc}}_{0:h}\mathbf{x}_{t-1}$. Let $\mathbf{z}_t$ be the high-level features $\mathbf{f}_{0:h}(\mathbf{x}_t)$, and let $s(\mathbf{x}, \mathbf{y})$ be the cosine similarity between vectors $\mathbf{x}$ and $\mathbf{y}$ in the same latent space. Our full loss over a batch is subsequently

$$\mathcal{L} = \sum_{i=1}^N \mathcal{L}_{\text{matr}}(\mathbf{x}_t^{(i)}) + \alpha\mathcal{L}_{\text{contr}},$$
$$\mathcal{L}_{\text{contr}} = -\frac{1}{N}\sum_{i=1}^N \log \frac{e^{s(\mathbf{z}_t^{(i)}, \mathbf{z}_{t-1}^{(i)})}}{\sum_{j=1}^N e^{s(\mathbf{z}_t^{(i)}, \mathbf{z}_{t-1}^{(j)})}} - \frac{1}{N}\sum_{j=1}^N \log \frac{e^{s(\mathbf{z}_{t-1}^{(j)}, \mathbf{z}_t^{(j)})}}{\sum_{i=1}^N e^{s(\mathbf{z}_{t-1}^{(i)}, \mathbf{z}_t^{(j)})}},$$

where N is our batch size and $\mathbf{x}_t^{(i)}, \mathbf{z}_t^{(i)}$ are the ith model activation and SAE latent vector in the batch, respectively. In practice, we load activations in pairs $\mathbf{x}_t, \mathbf{x}_{t-1}$ and shuffle the pairs to get diversity in each batch. We additionally explore an approach where we sample the contrastive pair uniformly over past tokens $\mathbf{x}_1, ..., \mathbf{x}_{t-1}$ to encourage long-range semantic consistency (see Sec. 4.6).

While the contrastive loss is applied only to high-level features, for low-level, token-specific features, we do not apply any constraints. By nature of fitting the residual data left unexplained by the high-level component of the network, our loss naturally encourages the low-level latents to capture remaining, fluctuating features over a sequence.

4 Experimental Evaluation

In this section, we evaluate Temporal SAEs. In Sec. 4.2, we evaluate our Temporal SAE's recovery and disentanglement of semantic, contextual, and syntactic content, in Sec. 4.3, we report results on

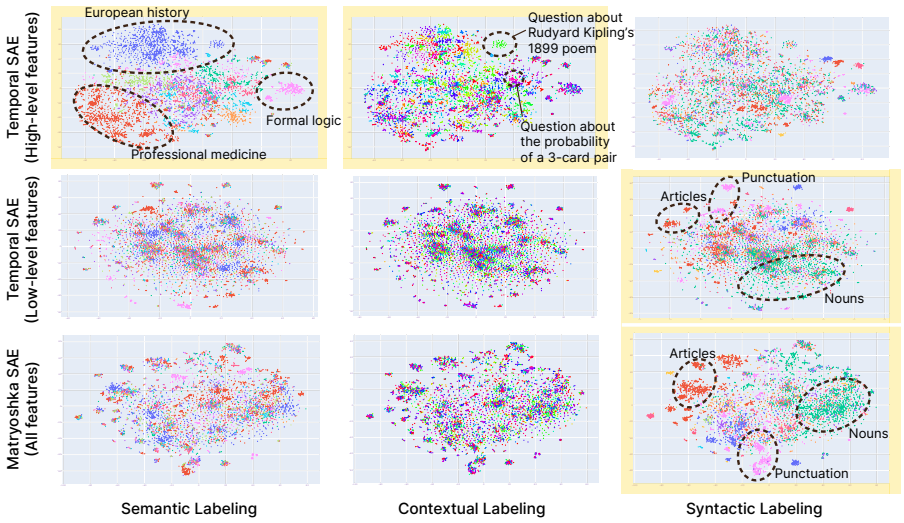

Figure 2: TSNE visualizations of Pythia-160m SAE decompositions of MMLU questions, labeled by question category (left), question number (middle column), and token part-of-speech (right). We see that the high-level features from T-SAEs (top) recover semantic and contextual information. The low-level features of T-SAEs (middle row), as well as Matryoshka SAEs (bottom), recover syntactic information.

standard SAE evaluation metrics, in Sec. 4.4, we measure various aspects of temporal consistency, in Sec. 4.5, we present a case study of how SAEs can help to uncover safety-relevant concepts and mechanisms in LLMs, and finally in Sec. 4.6, we provide results on various ablation studies.

4.1 Implementation Details

Models and Datasets. We conduct all experiments on Pythia-160m (Biderman et al., 2023) and Gemma2-2b (Team et al., 2024). We compare against baselines of BatchTopK SAEs (Bussmann et al., 2024), Matryoshka SAEs (Bussmann et al., 2025), and the model latents themselves when applicable. All models are trained and tested on the Pile (Gao et al., 2020). All probing evaluations are done on MMLU (Hendrycks et al., 2020), Wikipedia Wikipedia (2004), and FineFineWeb (M-A-P, 2024).

Hyperparameters. We train Pythia SAEs on layer 8 and Gemma SAEs on layer 12. All SAEs are trained with a batch-k-sparsity of 20, a dimensionality of 16k features, the BatchTopK activation, and the auxiliary loss from (Bussmann et al., 2025; Gao et al., 2024). These layers and hyperparameters are chosen to allow for comparability with pretrained and evaluated SAEs on Neuronpedia (Lin, 2023). Temporal and Matryoshka SAEs are trained with 20%-80% feature splits, where for Temporal SAEs the 20% are the high-level features. We use a regularization parameter of 1.0 on the temporal loss for all Temporal SAEs. For ablations on hyperparameters, please see Section 4.6.

4.2 Probing for Semantics, Context, and Syntax

To understand the types of features that the temporal loss encourages SAEs to learn, we evaluate the ability of Temporal SAEs to recover different types of linguistic information, namely semantic, contextual, and syntactic. We provide qualitative visualizations of these results in Figure 2 and quantitative probing results in Figure 3.

In Figure 2, we present TSNE visualizations of the Pythia-160m SAE activations for various questions taken from the MMLU dataset, which we color by the semantic content of the question, the contextual information for each token, and the syntactic information. In particular, we encode 20 tokens from each question into the SAE's feature space, and use TSNE as a dimensionality reduction and visualization method to understand how the SAE embeddings are clustered. We use the ques-

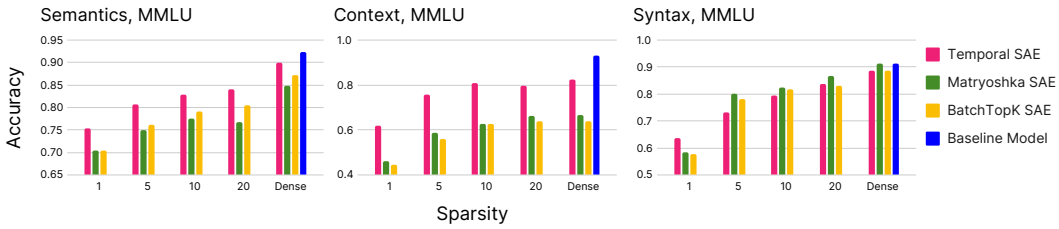

Figure 3: Accuracy of probes trained on SAE decompositions for various SAEs trained on Gemma2-2b, as well as probes trained directly on model latents (orange), with semantic labels (right), contextual labels (middle), and syntactic labels (right) with varying levels of probe sparsity (setup from (Kantamneni et al., 2025)). Dense probes are trained on all features.

tion category, such as "High School European History" or "Professional Medicine" as proxies for the semantic information. The contextual information simply refers to the question ID of each token, meaning tokens from the same question come from the same context and thus should have the same color. We use an open-source NLP library, spaCy (Honnibal et al., 2020), to retrieve part-of-speech labels for each token as a proxy for their syntactic content.

We find that the activations of high-level features from Temporal SAEs cluster strongly according to semantic content (top left) and contextual content (top middle), meaning tokens from the same question are represented similarly, as well as questions from the same topic. On the other hand, activations of the low-level features seem to be more syntactic, with stronger clusters for parts-of-speech (middle right). Matryoshka SAE embeddings prioritize syntactic information strongly over semantic and contextual information, with clear clustering for part-of-speech (bottom right) and minimal clustering for the other two labels.

Probing Validation. In order to validate these visual results, we probe the SAE activations for these same labels, using both k-sparse probing from Kantamneni et al. (2025) as well as normal Logistic Regression probes on the full activations. We train probes on $k = 1, 5, 10, 20$ features, where we select the features by comparing the mean activations of the positive and negative examples for each class from the train set. Note that dense probes trained on SAE activations have a dimensionality of 16k whereas dense probes trained directly on the model's residual stream have a dimensionality of 768 for Pythia-160m and 2304 for Gemma2-2b. We find that these results quantitatively reflect the qualitative results from the TSNE plots, with Temporal SAEs outperforming the baseline SAEs significantly for semantic and contextual labels, with little-to-no performance drop for syntactic information. Probing results for two more datasets, Wikipedia and FineFineWeb, as well as for Pythia-160m, are in Appendix A.2.2, with the same trends across all models and datasets.

Feature Disentanglement. While we see that Temporal SAEs recover more semantic and contextual information, can this behavior be attributed to the high-level features alone? Indeed, we observe specialization between high- and low-level features in Figure 2, where the high-level features exhibit semantic and contextual structure, whereas low-level features exhibit syntactic structure. To further characterize this behavior, we report probing accuracy on the each feature splits separately (see Appendix A.2.1), which confirms this same specialization. Interestingly enough, despite not performing the reconstruction task on their own due to the Matryoshka training loss, the low-level features are able to recover syntactic information. In contrast, for Matryoshka SAEs, across all tasks, performance can be attributed almost entirely to the high-level feature split, with the low-level features being significantly less predictive for semantics, context, and syntax, indicating a lack of disentanglement of high- and low-level features in the baseline Matryoshka SAEs.

4.3 SAE Evaluation

In Table 1, we provide results for various standard evaluations of SAEs to ensure that temporal consistency does not significantly degrade reconstruction quality. We define our metrics as such. **Fraction Variance Explained (FVE):** The fraction of the SAE input data's total variance that is successfully captured by the SAE decomposition. **Cosine Similarity (Cos Sim):** The co-

sine similarity between the SAE inputs and outputs. **Fraction Alive:** The proportion of SAE features that activate at least once on the test data. **Activation Smoothness:** For each sequence s, we filter for active features (features that fire at least once over the sequence). We compute $\Delta_s = \frac{1}{n'} \sum_{i=1}^{n'} \max_{t \in [1 \ldots T]} |\mathbf{f}_i(\mathbf{x}_t) - \mathbf{f}_i(\mathbf{x}_{t-1})| / \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2$ the average max absolute change over the n' active features normalized by the change in model latents[3]. Finally, we average over multiple sequences to get a smoothness score, $S = \frac{1}{n} \sum_s^n \Delta_s$. **Automated Interpretability (Autointerp) Score:** Score for how correct feature explanations are. We use SAEBench (Karvonen et al., 2025) to generate and score feature explanations with Llama3.3-70B-Instruct. For each latent, the LLM generates potential feature explanations based on a range of activating examples. Then, we collect activating and non-activating examples and ask a judge (also Llama3.3-70B-Instruct) to use the feature explanation to categorize examples and score its performance.[4]

We see that Temporal SAEs perform nearly equivalently to both Matryoshka and BatchTopK SAEs for both Pythia-160m and Gemma2-2b for FVE, Cosine Similarity, Fraction of Alive Features, and Autointerpretability score, meaning the added improvement in temporal consistency and semantic information recovery does not come at a significant cost to core SAE performance.

Table 1: Core Performance Metrics. We report smoothness on the high-level split (H) and standard deviations for autointerpretability scores.

		FVE (↑)	Cos Sim (↑)	Fraction Alive (↑)	Activation Smoothness (↓)	Autointerp Score (↑)
Pythia 160m	Temporal SAE	0.94	0.93	0.87	0.09 (H)	0.81 ± 0.17
	Matryoshka SAE	0.95	0.94	0.89	0.12	0.83 ± 0.16
	BatchTopKSAE	0.95	0.94	0.84	0.13	0.85 ± 0.15
Gemma 2-2b	Temporal SAE	0.75	0.88	0.78	0.10 (H)	0.83 ± 0.15
	Matryoshka SAE	0.75	0.89	0.76	0.14	0.83 ± 0.16
	BatchTopKSAE	0.76	0.89	0.66	0.13	0.83 ± 0.16

4.4 Temporal Consistency

Table 1 indicates that Temporal SAE features are smoother and more consistent than baselines. To explore this further, we visualize the top feature activations of Temporal SAEs over long sequences of text (Figs. 1, 4). In Figure 4, we concatenate four sequences of text: a biology question (MMLU), a letter from Charles Darwin (Project Gutenberg), an article on Animal Farm (Wikipedia), and a mathematics question (MMLU), and interpret them with a Temporal SAE. We take the top-8 most active features across the sequence and plot their activations for each token. The Temporal SAE features have clear phase transitions between texts, with different features activating and deactivating for specific sequences. Furthermore, within a sequence, these features are relatively smooth, without spiky, high-frequency changes in activation from token to token. We can even detect periodic behavior in the biology question, corresponding to each multiple-choice answer option. We note that there is an interesting leakage of features that continue to fire in later, semantically unrelated sections of the spliced sequence, highlighting that the model may retain past context and that the Temporal SAEs are able to identify this information rollover. Figure 1 conducts a similar analysis over Newton's Principia (Project Gutenberg), a genetics question (MMLU), and the Bhagavat Gita (Project Gutenberg) and compares the top features provided by the Temporal SAE to those generated by a baseline Matryoshka SAE. We find similar consistency and smoothness over sequences with clear transitions between sequences for the Temporal SAE, whereas the top Matryoshka SAE features activate across all three sequences without differentiating them, and are much noisier and high-frequency, fluctuating across tokens. Finally, we interpret these features using automated interpretability and report their explanations in the figure as annotations about the activations. We find that the labels reflect the true underlying semantic content present in each component sequence, with the 'Animal Farm' Wikipedia article activating a feature for "`Historical literature`

[3] fThis can be viewed as the average per-feature Lipschitz constant across sequences.

[4] Note that both the generation and scoring phase are highly noisy and dependent on the LLM judge.

and academic writing" in Figure 4 and the Bhagavat Gita excerpt in Figure 1 activating a "Worship and spiritual practices" feature.

Sequence-level Interpretability. Figure 1 and prior work (Sun et al., 2025) demonstrate how baseline SAE features are "dense," in that they fluctuate frequently over a sequence. In doing so, they can only provide human-interpretable explanations at the token level; as soon as you zoom out, you get an overwhelming set of activating features with very little structure or parseability. The smoothness of Temporal SAEs unlock a sequence-level understanding of data which was previously much harder, if not impossible, to parse from baseline SAE explanations (Figs. 1, 4).

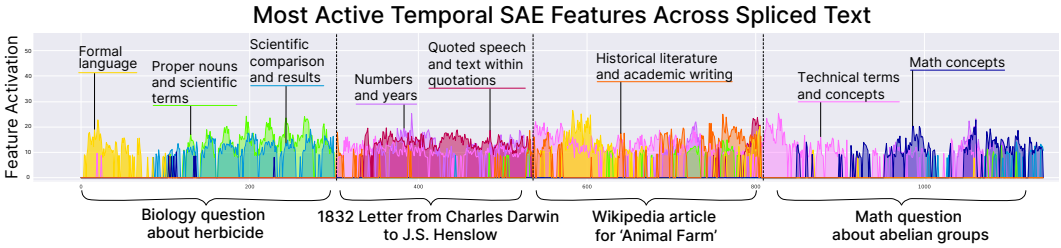

Figure 4: Top 8 most active Gemma2-2b Temporal SAE features over a concatenated sequence of text. Temporal SAE features exhibit clear phase transitions between sequences, are relatively smooth, and have explanations relevant to the semantic content of each component sequence.

4.5 A Case Study in Using Temporal SAEs to Understand Safety Labeling

For many safety applications where we know what we are trying to measure and avoid, it's not necessarily clear that unsupervised concept discovery methods are more effective than supervised methods, such as probing, steering, or finetuning. However, unsupervised methods can be an incredibly helpful tool for surfacing spurious correlations as well as unforeseen failure modes and vulnerabilities. To illustrate this, we conduct a case study using Temporal SAEs to examine how models represent and process safety-related concepts.

In particular, we convert the k-sparse probing method into a concept bottleneck approach (Koh et al., 2020) to investigate safety-related concept learning. Using the PKU-Alignment Beavertails Dataset (Ji et al., 2023), we train sparse probes on a binary safety task, where unsafe data is sampled from the subcategories drug_abuse/weapons/banned_substance, sexually_explicit/adult_content, and privacy_violation. We set $k = 32$ and our probe reaches a test AUC of 0.726 compared to a baseline AUC of 0.746 for probing the base model latent. This allows us to then create a pathway from model representations to safety classifications, where SAE features serve as human-interpretable concept bottlenecks that explain the model's safety assessments.

Our analysis reveals that the Temporal SAE features most predictive of unsafe content align remarkably well with intuitive safety concerns. Four of the top five predictive features for unsafe text include: "disease transmission methods," "words related to politics and social issues," "cybersecurity and hacking concepts," and "erotic content," all topics we would expect a safety classifier to flag as potentially problematic. Equally revealing are the features most predictive of safe content. Three of the top five concepts that correlate with safe text classifications include: "concepts of purpose and values," "government laws and regulations," and "health benefits." This suggests that models may associate content discussing ethical frameworks, legal compliance, and positive outcomes with safety. The prominence of these concepts as safety indicators suggests that adversarial prompts could potentially exploit these associations by framing harmful requests within contexts that superficially appear to discuss ethics or legal compliance.

4.6 Ablation Studies

In the following ablation studies (Table 2), we explore the effect of varying components in our Temporal SAE training pipeline. All results are reported as the difference between the ablation and

Table 2: Ablation study of training configurations for Pythia-160m Temporal SAEs.

	FVE	Fraction Alive	Activation Smoothness (H)	Semantics	Context	Syntax
Random Contrast	0.0	-0.05	0.0	-0.02	+0.11	-0.10
50:50 Split	-0.01	+0.01	0.0	+0.02	+0.09	-0.08
10:90 Split	0.0	-0.07	-0.02	-0.01	+0.01	+0.01
No Contrastive	+0.01	+0.06	+0.07	-0.07	-0.1	+0.01

the normal Temporal SAE with the implementation described in Section 4.1. We conduct semantics, context, and syntax sparse probing evaluations on MMLU with $k = 5$. First, we conduct a hyperparameter sweep on the high- and low-level feature percentages, varying the proportions to 10:90 and 50:50. As expected, if we increase the size of our high-level split, Temporal SAEs better recover semantics and context but perform worse on syntax. We next study the impact of contrasting with a token sampled randomly from any previous token in the context window, the $t - r$th token with $r < 25$ rather than from the $t - 1$st token. When contrasting with a random token, we incorporate even longer-range dependencies into our temporal constraint; as a result of this, we see a large increase in performance on the context task and a large decrease in performance on the syntax task, but with minor change to semantic performance. Depending on the interpretability application, this behavior may be preferable to that of the Temporal SAEs trained on the immediate previous token, highlighting the need to carefully consider the features we hope to find when using unsupervised concept discovery methods. Finally, we explore the impact of the contrastive component of our loss term, and consider the naive baseline of a sample-wise temporal similarity loss, $\ell_i = \alpha\|\mathbf{z}_t^{(i)} - \mathbf{z}_{t-1}^{(i)}\|_2^2$, averaged over a batch. This naive approach enforces less structure in the high-level feature space, resulting in worse semantic and contextual performance but allowing for better performance on standard reconstruction metrics.

5 Discussion and Conclusions

The efficacy of "bottom-up" interpretability methods, which aim to discover and represent concepts learned by large neural networks in an unsupervised fashion, has been a fiercely contested topic. In recent years, dictionary learning methods such as Sparse Autoencoders were hailed as a triumph in the interpretability community, showing promise in their ability to uncover unexpected and novel concepts, and providing a potential path forward for steering and control. However, as the excitement wore off, it became clear that SAEs suffer from a variety of issues, one of which being that they systematically fail to capture the rich conceptual information that drives linguistic understanding, instead exhibiting a bias towards shallow, syntactic features, such as ``the phrase 'The' at the start of sentences.'' This lack of deeper semantic concepts could potentially be attributed to the underlying LLMs themselves; maybe they don't truly understand semantics or pragmatics, and it's thus unrealistic to expect that interpretability methods would find them. But more likely than not, current concept discovery methods simply aren't good enough to reveal the types of features we generally are interested in and that LLMs likely encode.

In this work, we propose that this is due to a fundamental issue with how dictionary learning methods for LLMs are trained. Language itself has a rich, well-studied structure spanning syntax, semantics, and pragmatics; however, current unsupervised methods largely ignore this linguistic knowledge, leading to poor feature discovery that favors superficial patterns over meaningful concepts. We focus on a simple but important aspect of language: semantic content has long-range dependencies and tends to be smooth over a sequence, whereas syntactic information is much more local. We propose a novel loss function for training SAEs such that a subset of features behaves smoothly over time, better extracting semantic features from data. Through experiments, we demonstrate that the features learned by Temporal SAEs are more semantically structured, with minimal loss to reconstruction performance. We also present a case study demonstrating how this semantic information can be valuable in practical, real-world applications, such as finding safety vulnerabilities.

6 Reproducibility Statement

All language models, libraries, and datasets used in this paper are publicly available and open-source. We describe our implementation in our section on Implementation Details (Sec. 4.1), our training approach and model architecture in detail in Section 4.4, and highlight the existing libraries and methods we use for evaluation in Section 4.3. We also describe our custom evaluations in detail in the experiments section 4. Upon publication, we will open-source our codebase and T-SAEs for further reproducibility.

7 Ethics Statement

Interpretability is closely tied to the ethical development and application of AI systems. On one hand, developing better understanding of models can help highlight biases and failure modes to ensure they treat all people ethically. On the other hand, the understanding they bring can allow malicious actors to more efficiently jailbreak models or control them for nefarious purposes. We are aware of the potential dual uses of such a technique but hope that improving the transparency of AI systems will enable the development of safer AI systems.

References

Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. Towards unifying interpretability and control: Evaluation via intervention, 2025. URL https://arxiv.org/abs/2411.04430.

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*, pp. 2397–2430. PMLR, 2023.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Decomposing language models with dictionary learning. 2023.

Roger Brown. *A first language: The early stages*. Harvard University Press, 1973.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT press, 1965.

Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

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

Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

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.

Thomas Griffiths, Mark Steyvers, David Blei, and Joshua Tenenbaum. Integrating topics and syntax. *Advances in neural information processing systems*, 17, 2004.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Sai Sumedh R Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. *arXiv preprint arXiv:2503.01822*, 2025.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi: 10.5281/zenodo.1212303.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, et al. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

Johnny Lin. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. URL `https://www.neuronpedia.org`. Software available from neuronpedia.org.

Weixin Luo, Wen Liu, Dongze Lian, Jinhui Tang, Lixin Duan, Xi Peng, and Shenghua Gao. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):1070–1084, 2019.

Xinrun Du* Zhimiao Yu* Zili Wang* Zekun Wang Shuyue Guo Tianyu Zheng Kang Zhu Jerry Liu Shawn Yue Binbin Liu Zhongyuan Peng Yifan Yao Jack Yang Ziming Li Bingni Zhang Minghao Liu Tianyu Liu Yang Gao Wenhu Chen Xiaohuan Zhou Qian Liu Taifeng Wang+ Wenhao Huang+ M-A-P, Ge Zhang*. Finefineweb: A comprehensive study on fine-grained domain web corpus, December 2024. URL `[https://huggingface.co/datasets/m-a-p/FineFineWeb](https://huggingface.co/datasets/m-a-p/FineFineWeb)`.

Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

Helen J Neville, Debra L Mills, and Donald S Lawson. Fractionating language: Different neural subsystems with different sensitive periods. *Cerebral cortex*, 2(3):244–258, 1992.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Frank Ong and Michael Lustig. Beyond low rank+ sparse: Multiscale low rank matrix decomposition. *IEEE journal of selected topics in signal processing*, 10(4):672–687, 2016.

Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability. *arXiv preprint arXiv:2501.18823*, 2025.

Xiaoqing Sun, Alessandro Stolfo, Joshua Engels, Ben Wu, Senthooran Rajamanoharan, Mrinmaya Sachan, and Max Tegmark. Dense sae latents are features, not bugs. *arXiv preprint arXiv:2506.15679*, 2025.

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.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Tiep Huu Vu and Vishal Monga. Fast low-rank shared dictionary learning for image classification. *IEEE Transactions on Image Processing*, 26(11):5160–5175, 2017.

Wikipedia. Wikipedia, the free encyclopedia, 2004. URL https://en.wikipedia.org.

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

Mingfang Zhang, Jarod Lévy, Stéphane d'Ascoli, Jérémy Rapin, F Alario, Pierre Bourdillon, Svetlana Pinet, Jean-Rémi King, et al. From thought to action: How a hierarchy of neural dynamics supports language production. *arXiv preprint arXiv:2502.07429*, 2025.

A Appendix

A.1 Table of Contents

A.2 Additional Results

In the following sections, we detail additional results across splits for Temporal SAEs and Matryoshka SAEs, as well as on additional datasets.

A.2.1 Metrics across high and low splits

In Figure 5 and Table 3, we report probing and benchmark results across feature splits for Temporal SAEs and Matryoshka SAEs. We find that the high-level Temporal SAE feature space is smoother and contains more semantic and contextual information than its low-level. Additionally, we find syntactic information is equally spread between high- and low-level features due to the Matryoshka training setup (see discussion on disentanglement in Section 4.2). In contrast, Matryoshka SAEs are less smooth and place most syntactical information in the high-level split and are worse at semantics and context tasks across splits.

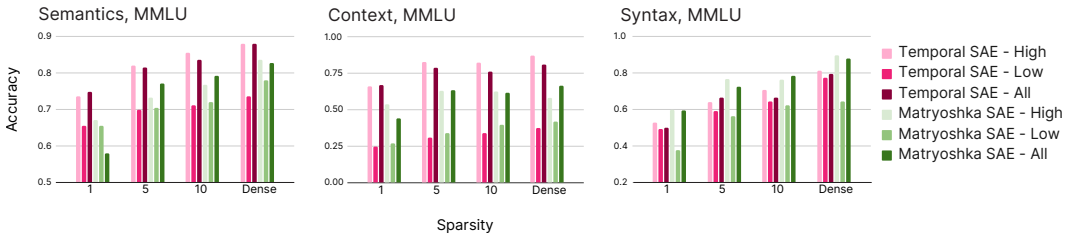

Figure 5: Probing results split across high and low splits for Temporal and Matryoshka SAEs.

Table 3: SAE Benchmarks split across high and low splits, when applicable.

		Activation Smoothness	High	Low
Pythia 160m	Temporal SAE	0.12	0.09	0.17
	Matryoshka SAE	0.12	0.12	0.13
	BatchTopKSAE	0.13	-	-
Gemma 2-2b	Temporal SAE	0.13	0.10	0.15
	Matryoshka SAE	0.14	0.15	0.12
	BatchTopKSAE	0.13	-	-

A.2.2 Probing Results on More Datasets

In the following plots, we report semantic, contextual, and syntactic probing results on both Gemma (Fig. 6) and Pythia (Fig. 7) Temporal SAEs for the FineFineWeb, MMLU, and Wikipedia datasets. Results for Gemma on MMLU are in the main paper (Fig. 3).

A.2.3 More TSNE Results

In Figure 8, we present TSNE visualizations of the Pythia-160m SAE activations for more baseline methods than shown in Figure 2. Please see Section 4.2 for more information.

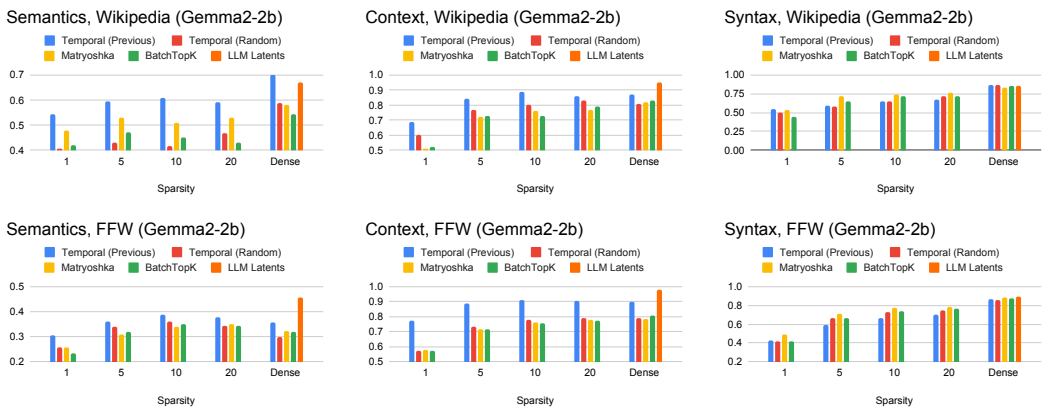

Figure 6: Accuracy of probes trained on SAE decompositions of Wikipedia and FineFineWeb for various SAEs trained on Gemma2-2b, as well as probes trained directly on model latents (orange), with semantic labels (right), contextual labels (middle), and syntactic labels (right) with varying levels of probe sparsity (setup from Kantamneni et al. (2025)). Dense probes are trained on all features.

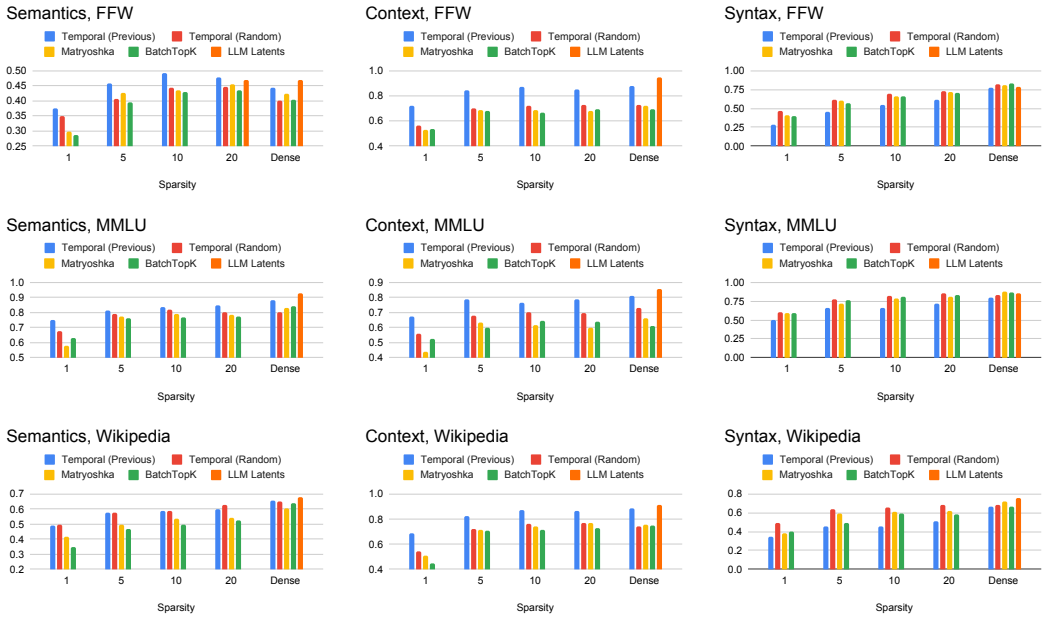

Figure 7: Accuracy of probes trained on SAE decompositions of data from FineFineWeb (top), MMLU (middle), and Wikipedia (bottom) for various SAEs trained on Pythia-160m, as well as probes trained directly on model latents (orange), with semantic labels (right), contextual labels (middle), and syntactic labels (right) with varying levels of probe sparsity (setup from Kantamneni et al. (2025)). Dense probes are trained on all features.

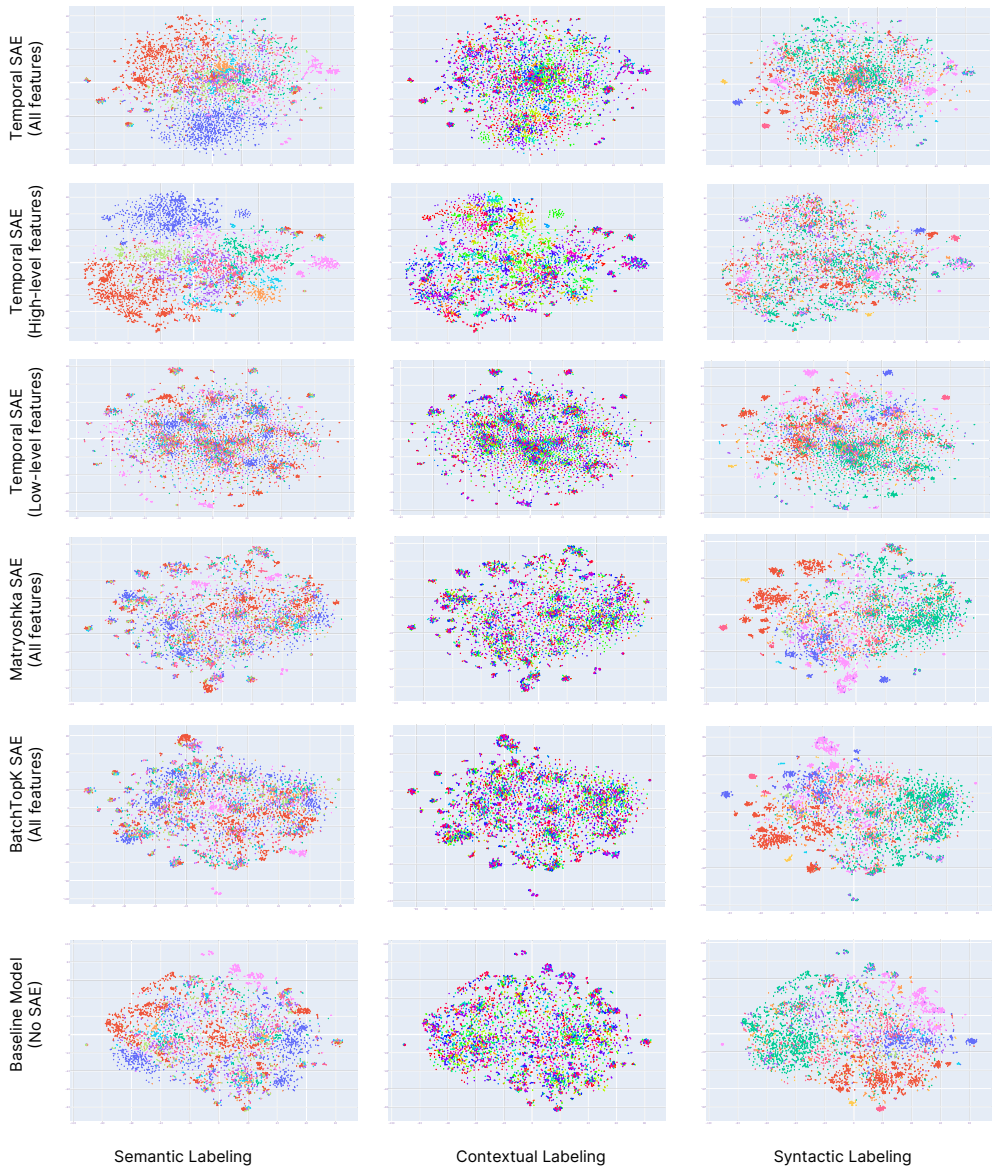

Figure 8: TSNE visualizations of Pythia-160m SAE decompositions of MMLU questions, labeled by question category (left), question number (middle column), and token part of speech (right). We see that the high-level features from Temporal SAEs (second row) recover semantic and contextual information. The low-level features of Temporal SAEs (third row), as well as Matryoshka and BatchTopK SAEs (fourth and fifth row), recover syntactic information. Baseline model latent (last row) balance a mix of information.