Interpretable Sparse Features for Probing Self-Supervised Speech Models

Iñigo Parra
University of California, Berkeley
Department of Linguistics
iparra@berkeley.edu

Abstract

Self-supervised speech models have demonstrated the ability to learn rich acoustic representations. However, interpreting which specific phonological or acoustic features these models leverage within their highly polysemantic activations remains challenging. In this paper, we propose a straightforward and unsupervised probing method for model interpretability. We extract the activations from the final MLP layer of a pretrained HuBERT model and train a sparse autoencoder (SAE) using dictionary learning techniques to generate an overcomplete set of latent representations. Analyzing these latent codes, we observe that a small subset of high-variance units consistently aligns with phonetic events, suggesting their potential utility as interpretable acoustic detectors. Our proposed method does not require labeled data beyond raw audio, providing a lightweight and accessible tool to gain insights into the internal workings of self-supervised speech models.

1 Introduction

Recent advances in self-supervised learning have produced speech models whose hidden representations support a wide range of downstream tasks without fine-tuning (Hsu et al., 2021; Baevski et al., 2020a; Chen et al., 2022). However, these models remain largely "black boxes": it remains unclear precisely which acoustic and linguistic aspects of the input signal are captured by individual layers or units. This lack of interpretability poses significant challenges for both theoretical understanding and practical applications, limiting our ability to effectively control, edit, or explain model outputs. Consequently, developing methods that show and inspect the internal workings of self-supervised models is an essential step toward more transparent and flexible speech technologies.

Prior approaches to probing the internal representations of self-supervised speech models have usually involved supervised classifiers trained to predict explicit phonetic or prosodic labels from hidden embeddings. Alternative methods have used linear projection techniques, such as principal component analysis (PCA) and canonical correlation analysis (CCA), to identify correlations between learned embeddings and linguistic categories (Martin et al., 2023; Pasad et al., 2021, 2024). While these studies demonstrate that self-supervised features correlate strongly with traditional linguistic categories, they do not yield interpretable, temporally aligned, discrete signals (Pasad et al., 2024; Gimeno-Gómez et al., 2025). Thus, they fall short of providing the detailed unit-level insights necessary for granular analysis or intervention.

In parallel, computational neuroscience has explored sparse coding models extensively, particularly emphasizing the emergence of discrete, interpretable "spiking" events. Such sparse representations often naturally align with salient perceptual phenomena and sensory boundaries in a human-readable format, making them particularly promising for probing complex activation patterns.

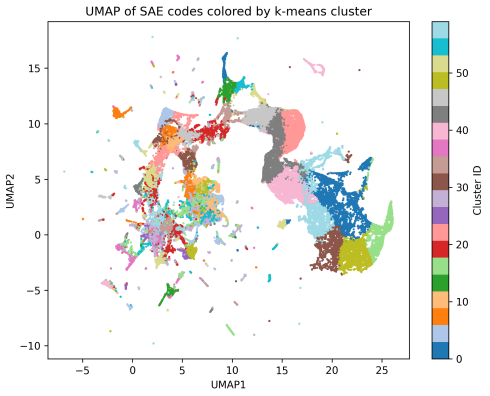

Figure 1: UMAP of a subset (10%) of TIMIT sparse representations. These were obtained after sparse-encoding the original 1024 dimensional MLP activations from HuBERT's last layer.

Proceedings of the IJCNLP-AACL 2025 Student Research Workshop, pages 1–9

Motivated by these insights, we introduce a sparse autoencoder (SAE) probe specifically designed to analyze self-supervised speech models. Our approach consists of three primary steps: (1) extracting the activations from the final feed-forward multilayer perceptron (MLP) layer of a frozen HuBERT model (`facebook/hubert-large-ls960-ft`[1]), yielding an activation matrix of dimensions (N_{frames}, D); (2) training a lightweight SAE (linear encoder projecting activations to an over-complete latent space; set to $4 \times D$ dimensions), enforced by an $L1$ sparsity penalty, and decoding back to the original dimensionality; and (3) performing analyses on the resulting sparse latent representations, including ranking latent units by variance, visualizing their temporal firing patterns, conducting k-means clustering, and embedding with uniform manifold approximation and projection (UMAP).

Our contributions are as follows:

- We propose a straightforward, unsupervised probing pipeline using sparse autoencoders to dissect and interpret the latent structure within pretrained HuBERT activations.
- We introduce the Q-SAE, a variant of the sparse autoencoder that incorporates a controllable low-dimensional continuous vector for enhanced interpretability and control.
- We demonstrate that high-variance sparse units behave analogously to neural "feature detectors", exhibiting discrete spiking behaviors.
- We provide our code for extraction, SAE training, and analysis, facilitating future research aimed at interpretability and controllability in self-supervised speech representations.[2]

The remainder of this paper is organized as follows. Section 2 comments on related work on speech representation probing, sparse coding methodologies, and their intersections. Section 3 outlines our proposed architectures, training procedures, and analytic methods in detail. Section 4 presents qualitative and quantitative analyses of the learned sparse codes. Section 5 situates these results within a broader theoretical and applied context. Finally, section 6 concludes by summarizing key insights and outlining limitations and potential directions for future research.

[1]The model is openly available at Hugging Face.

[2]All materials available upon acceptance.

2 Previous Work

Self-Supervised Speech Representations. Recent years have seen rapid progress in self-supervised learning for speech. Early models such as Wav2Vec (Baevski et al., 2020a) and its successor Wav2Vec 2.0 (Baevski et al., 2020b) learn frame-level latent embeddings by masking and contrastive predictive coding. HuBERT (Hsu et al., 2021) improved on these methods by iteratively clustering acoustic features and using cluster assignments as targets, yielding representations that match or exceed fully supervised baselines on phoneme recognition. More recently, Data2Vec (Baevski et al., 2022) unified self-supervised learning across modalities by predicting contextualized representations rather than discrete units.

These models improved downstream performance on speech recognition, speaker identification, and emotion detection tasks. Still, their internal activation patterns remain largely opaque.

Probing and Representation Analysis. To understand the internal mechanisms of models, previous work applied supervised probes and linear analysis techniques. Initially, the probes were used in text-based models such as BERT (Tenney et al., 2019). Linear probings demonstrated that models are able to capture different aspects of language in different layer depths (Tenney et al., 2019) or even individual attention heads (Clark et al., 2019).

Phonetic and prosodic probes train lightweight classifiers on frozen embeddings to predict linguistic labels (Pimentel et al., 2020; English et al., 2022). While such probes quantify which layers correlate with specific features, they require annotated data and only provide coarse-grained, timestep-agnostic scores. Unsupervised methods like PCA, CCA, and SVCCA examine subspace overlap between model layers (Raghu et al., 2017; Morcos et al., 2018), revealing global geometric structure but lacking temporal resolution. Information-theoretic measures, such as mutual information (MI) between representations and phonetic sequences, further characterize feature encoding but depend on explicit alignment (Pimentel et al., 2020).

Sparse Coding and Autoencoders. Sparse coding offers an alternative framework for discovering

interpretable, monosematic features. Seminal work showed that enforcing sparsity on natural images yields Gabor-like filters similar to early visual cortex (Olshausen and Field, 1996).

In deep learning, mainly in the textual modality, sparse representations have been used for dictionary learning (Bricken et al., 2023; Templeton et al., 2024). Sparse autoencoders allow to do this combining an encoder-decoder architecture with an L_1 penalty or KL-divergence constraint on the bottleneck (Ng et al., 2011), encouraging a small subset of active units per input. Such models can learn event-like activations without explicit supervision.

Clustering and Manifold Visualization. Clustering learned codes provided a direct view of emerging categories. K-means has long been applied to embeddings for unsupervised phoneme and speaker clustering (MacQueen, 1967). Modern work on self-supervised speech also leverages k-means, both within HuBERT's iterative clustering loop (Hsu et al., 2021) and as a post-hoc analysis tool (Baevski et al., 2020a). To visualize high-dimensional codes, techniques such as t-SNE and UMAP reveal salient manifold structure (McInnes et al., 2018), enabling qualitative assessment of category separation.

Interpretability in Time. Few studies achieve time-aligned, unit-level interpretability in self-supervised speech models. Most probes aggregate over time or collapse sequences to fixed vectors, obscuring dynamic events like phoneme boundaries or burst onsets. Sparse autoencoders can produce firing patterns that align with salient acoustic transitions.

To our knowledge, no prior work applies sparse encoding directly to HuBERT's (or any other speech model's) internal MLP activations to extract interpretable, monosemantic features. We have no knowledge of the Q-SAE being applied in previous work, where the main objective of the model is providing a low-dimensional vector to manipulate the monosemantic, sparse, feature space.

3 Methodology

3.1 HuBERT Activations

We analyze activations extracted from HuBERT (Hsu et al., 2021) (see Appendix A for model details) during inference on the TIMIT (Garofolo et al., 1993) dataset (see Appendix B for dataset information). HuBERT takes raw audio waveforms and outputs embedding representations which correspond to 20ms frames (16kHz). An initial CNN waveform encoder creates audio patches, which are processed by a transformer encoder (BERT-like; trained on masked token prediction). The patches are linearly projected to obtain the embedding representations that approximate discrete phonetic units.

As in previous work in the text modality (Bricken et al., 2023; Templeton et al., 2024), we analyze the MLP activations from HuBERT's last layer. We extract the activation using a forward hook during inference on the training split of TIMIT. For each waveform, we obtained 1024-dimensional activation vectors of n frames. We collapsed batch and n dimensions to form a dataset with shape $N \times 1024$, where N are the total activation examples ($N = 762,438$).

3.2 Models

We propose two architectures to extract sparse features from dense activation vectors: a Sparse Autoencoder (SAE) and the Q-Autoencoder (Q-SAE). We trained both architectures with dictionary learning purposes.

3.2.1 Sparse Autoencoder

Architecture. The SAE follows a vanilla implementation (Figure 2), where the input sequence x is mapped into an over-complete latent space z, and is later reconstructed into $\hat{x}$. The encoder is encouraged to induce sparsity of z through an $L1$ penalty included in the optimization objective. The decoder has to map the sparse representations back to the original input.

Optimization Objective. The objective is defined as a dual cost function with a tunable parameter λ on the sparsity penalty:

$$\mathcal{L}_{\text{SAE}} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}_{\text{MSE Reconstruction}} + \overbrace{\lambda \cdot ||z||_1}^{L1 \text{ Sparsity}}.$$

The first term forces the model to reconstruct the input data as faithfully as possible while the second forces the sparsity of features. The tuneable lambda parameter allows to control the level of sparsity of the over-complete latent space. Higher lambda values shrink the values to zero, while lower values preserve more activations. We measure the percentage of active units through L_0 and aim at a final value of $\approx 3\%$ active units.

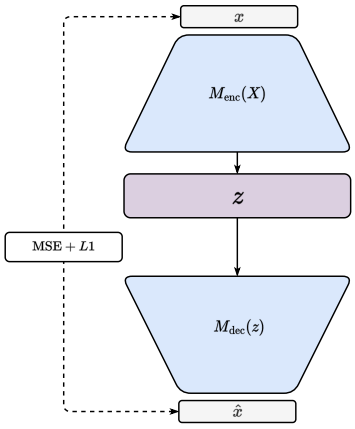

Figure 2: Sparse Autoencoder architecture.

3.2.2 The Q-Sparse Autoencoder

Architecture. The Q-SAE follows a similar architecture with additional components that allow a general control over features in space z. We followed a SAE architecture with the addition of a Q-Net (Chen et al., 2016), a continuous vector c, and a top-k feature selector mechanism on z (Figure 3). As in the SAE, an input sequence x is mapped into a sparse representation z.

Top-k Mechanism. In this variant, we apply a feature selector function $\text{Topk}(\cdot)$ on z, which constraints the decoder to access only the top-k most prominent features in z. For a single latent vector $z \in \mathbb{R}^D$, the mechanism is defined as follows. Let $k = \max(1, \lfloor k_{\text{frac}} \cdot D \rfloor)$ and $S \subset \{1, ..., D\}$ be the set of indices of the k entries of z with largest absolute value. Then, the top-k operator is defined as

$$[\text{Topk}(z)]_j = z_j \cdot \left[\mathbf{1}_{\{j \in S\}} = \begin{cases} z_j, & \text{if } j \in S \\ 0, & \text{if } j \notin S \end{cases} \right]$$

where $\mathbf{1}_{\{\cdot\}}$ is a masking operator.

Continuous Vector c. After the selection step, a continuous vector $c \sim \mathcal{N}(0, 1)$ is concatenated to the resulting latent space $\text{Topk}(z)$. The decoder takes the concatenated representation as input and outputs a reconstruction $\hat{x}$. The output is further fed into the Q-net and is encouraged to predict the continuous vector c. In this way, the decoder is forced to rely on the sparse representation $\text{Topk}(z)$ and the continuous vector c to reconstruct the input sequence.

Optimization Objective. The objective of the Q-SAE is similar to that of the SAE: the model is encouraged to reconstruct the input data x from a sparse representation z. In the Q-SAE, the most prominent features of z are selected through the top-k selector, which acts on on z with the purpose of passing only meaningful sparse features to the decoder. In addition, a continuous vector c is concatenated to the filtered z space, which is processed by the decoder to predict $\hat{x}$.

The support Q-net predicts a continuous vector $\hat{c}$ from $\hat{x}$ and is optimized using a mutual information (MI) cost function to encourage c to include meaningful information about x. This forces c to be used during decoding, so that we can later use low-dimensional continuous vectors to modify relevant features of z. The final objective is defined as

$$\mathcal{L}_{\text{SAE}} = \overbrace{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}^{\text{MSE Reconstruction}} + \underbrace{\lambda \cdot ||\text{Topk}(z)||_1}_{L1 \text{ Sparsity}}$$

$$\mathcal{L}_{\text{Q}} = \underbrace{\beta \cdot \text{InfoNCE}(\hat{c}, c)}_{\text{MI}}$$

$$\mathcal{L}_{\text{Q-SAE}} = \mathcal{L}_{\text{SAE}} + \mathcal{L}_{\text{Q}}$$

where InfoNCE (Oord et al., 2018) is the contrastive loss function and MI term that pushes the Q-net's predictions $\hat{c}$ to be informative.

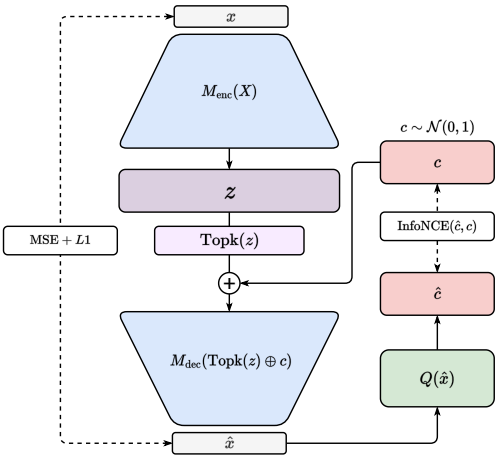

Figure 3: Q-Sparse Autoencoder architecture.

Data and Training. We train our models on a self-supervised regime using the activations extracted from HuBERT during inference on the TIMIT dataset.

After training both architectures, we choose the vanilla autoencoder for the following reasons. First, the feature disambiguation is more straightforward in the sense that it avoids an extra cost objective. Second, the original objective of the study is more

aligned with the central purpose of the vanilla SAE: disentangle polysemanticity. However, we propose the Q-SAE (or potential variants) as promising alternatives useful for causal interpretability.

Figure 4 shows three training runs of the SAE architecture with different λ values. Following previous work (Bricken et al., 2023; Templeton et al., 2024), we aimed at preserving 3% of active units in the latent space. We use the model trained with $\lambda = 0.09$ as our model for experimentation. Model selection was not mainly guided by a minimal test loss criterion, but rather as a mixed one giving preference to the model with best z space representations.

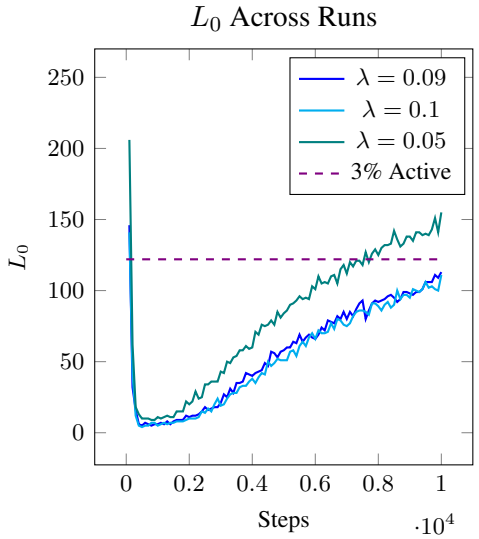

Figure 4: L_0 tracking of the SAE model across three runs with different sparsity lambda values. The dashed line indicates the 3% active units frontier.

Table 1 shows a high level summary of the training runs of each model. Following (Bricken et al., 2023; Templeton et al., 2024) we use a latent space four times the original input size.

4 Results

Sparse Features Capture Phonetic Events. To verify that individual sparse dimensions behave like discrete event detectors, we extracted the ten features with highest activation variance and plotted their supra-threshold spiking patterns in Figure 5.

These top features activated in distinct, temporally sparse bursts, consistent with a spiking code. Several of these sparse codes showed structured, bursty activation patterns rather than random or uniformly distributed firing, suggesting they responded to recurring patterns in the input. Some units fired densely in specific time ranges, potentially corresponding to phonetic or acoustic units, while others showed more distributed or selective patterns. These observations supported the hypothesis that individual sparse units serve as feature detectors, encoding meaningful substructures in the representation space.

Model	**SAE**$_1$	**SAE**$_2$	**SAE**$_3$
Epochs	10	10	10
Input	1024	1024	1024
Latent D	4096	4096	4096
Factor	4	4	4
Sparsity λ	0.09	0.1	0.05
Optimizer	Adam	Adam	Adam
Grad Clip	1.0	1.0	1.0
L1 Train	0.16	0.15	0.24
L1 Test	0.18	0.17	0.27
MSE Train	0.58	0.58	0.57
MSE Test	0.48	0.48	0.46

Table 1: Training parameters for each sparse autoencoder run.

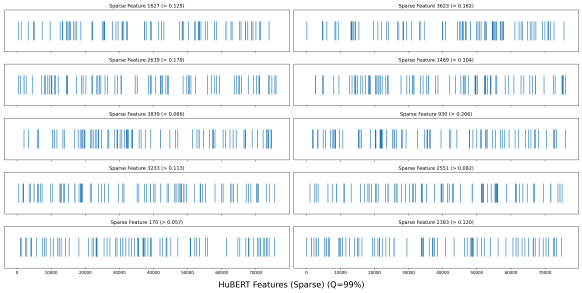

Figure 5: Temporal firing rasters of the top ten variance sparse features. Each panel shows the frame indices (x-axis) at which a given feature exceeds its 99th percentile threshold, revealing spike-like activations.

High-level clustering indicates "phonological hubs". To probe whether individual sparse dimensions acted like monosemantic feature detectors, we performed k-means clustering of the latent, over-complete, z representations. We show the most prominent phonological categories per cluster in Figure 6.

The heatmap analysis of phonological categories versus sparse code clusters indicated variability in how phonetic information was distributed across latent units. Clusters 22, 40, 42, and 57 show distinctly stronger associations with specific phonological categories, such as silence, vowels, and stops. This suggests that a subset of sparse codes preferentially encoded phonetic events more clearly

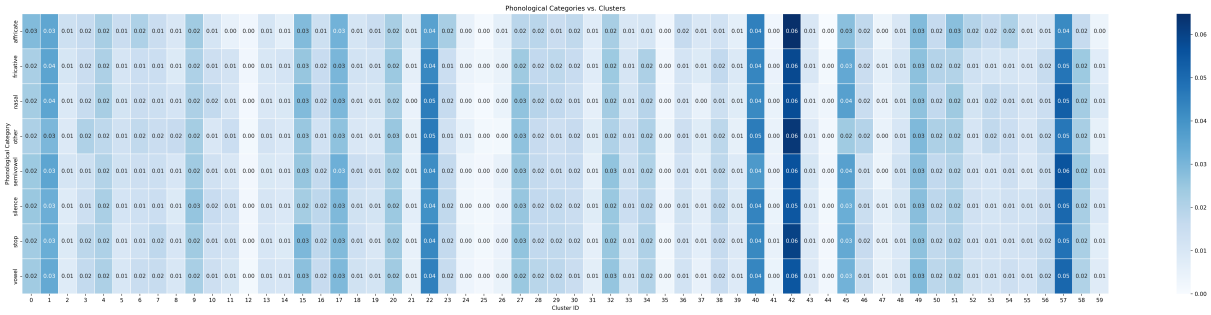

Figure 6: Confusion plot of the over-complete vectors z clusters vs phonological categories.

than others, highlighting their specialized role as potential feature detectors. In contrast, other clusters showed relatively uniform and lower activation levels across categories, underscoring the sparsity and selectivity of these high-variance units.

Features Can be Higher or Lower Order. We quantified the category specificity of each high-variance feature by averaging its activation over all frames of each phonological class (Figure 7).

Features 3233, 385, 2026, and 3623 showed a higher selectivity for affricates, while other dimensions yielded mean activations higher in stops than in vowels, indicating strong sensitivity to transient bursts and turbulence. Conversely, features such as 1627, 3320, and 170 activated across all categories, indicating polysemanticity. This indicated that the features were classified into low-order (including individualized category information) or high-order (detectors for various categories) selective classes.

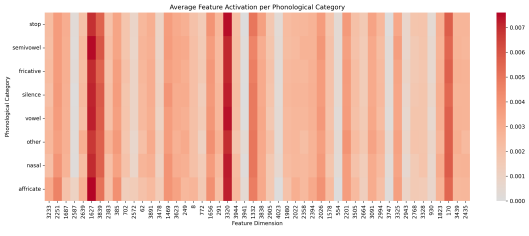

Figure 7: Average activation of the highest-variance sparse features, computed separately for each phonological category. Rows correspond to categories and columns to feature dimensions (sorted by variance).

5 Discussion

The primary objective of this study was to leverage sparse autoencoders (SAEs) as unsupervised probes for interpreting phonological information captured by self-supervised speech models, specifically HuBERT. Our findings underscore the efficacy of SAEs in uncovering discrete, phonetic events encoded within high-dimensional sparse spaces, highlighting their potential as powerful interpretability tools in speech processing.

One significant insight is the emergent nature of the sparse features extracted from the final MLP activations of HuBERT. High-variance sparse units align with phonetic units, suggesting these encode acoustic-phonetic events. This aligns well with classical phonetic theory, which emphasizes the acoustic saliency of such transitional points (Stevens, 2002). Low-variance units encode subtler phonetic nuances distributed across broader contexts, indicating a hierarchical structuring of phonological information within the latent space.

Another critical observation is the partial rather than complete monosemanticity of extracted features. Although some sparse units exhibit specificity towards particular phonetic events, many high-variance dimensions activate across multiple classes. This polysemanticity implies that the HuBERT model's internal representation inherently take advantage of phonetic information distributed across dimensions, a phenomenon consistent with previous findings in sparse coding research in other modalities (Bricken et al., 2023; Templeton et al., 2024). Consequently, future research might explore mechanisms to further disentangle these polysemantic representations, possibly via refined architectures or additional regularization techniques.

Additionally, our experimental results emphasize the limitations inherent to a purely unsupervised approach. While the sparse autoencoder provides valuable qualitative insights, interpreting the full phonetic scope of each unit's activations remains challenging without reference to external linguistic labels. A hybrid approach integrating sparse autoencoders with minimally supervised labeling or linguistic priors could enhance the interpretability and practical applicability of the proposed methodology.

The introduction of the Q-SAE, despite its intriguing potential for causal manipulation of sparse features through continuous vectors, requires further investigation. Our preliminary decision to favor the vanilla SAE was guided by simplicity and clearer interpretability. However, the Q-SAE's ability to manipulate sparse feature spaces via controllable vectors could significantly extend the framework's utility, especially in tasks requiring precise feature-level intervention, such as speech editing or targeted phoneme manipulation.

Finally, this study contributes methodologically by demonstrating the compatibility of sparse coding techniques, traditionally used in computational neuroscience, with contemporary deep learning models for speech. This intersection offers fertile ground for interdisciplinary research, potentially enabling cognitive insights into speech perception and informing the design of biologically inspired machine learning models.

Future work should focus on scaling this approach to larger and more diverse speech corpora, validating the robustness of our findings across languages and dialects. Additionally, exploring adaptive or dynamic sparsity constraints could refine the granularity of phonological features captured, further bridging computational techniques with linguistic theory.

6 Conclusion

We introduced an unsupervised probing pipeline that uses a sparse autoencoder to extract interpretable features from the final MLP activations of a pretrained HuBERT model. Our qualitative analyses show that: (i) high-variance latent units fire at linguistically meaningful phonetic events, and (ii) clustering those sparse codes recovers broad class groupings. These findings suggest that scaling the presented pipeline and sparse coding can uncover phonological structure in self-supervised speech models without any explicit supervision, providing a new tool for model interpretability and control.

Limitations

The performance of the models and the experimental results were heavily constrained by the available data. Further work should incorporate activations from different datasets and models to uncover potential universal behaviors across models. In addition, the study is limited to the analysis of one layer's MLP activations. Internal layers may yield more interpretable and comprehensive results. The Q-SAE is still under development, which posed a limitation to its usefulness for the case under study.

Ethics Statement

This work uses publicly available speech data and does not involve any personally identifiable or sensitive information. All analyses were performed on aggregate model activations.

References

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pages 1298–1312. PMLR.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020a. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Patrick Cormac English, John Kelleher, and Julie Carson-Berndsen. 2022. Domain-informed probing of wav2vec 2.0 embeddings for phonetic features. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 83–91.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

David Gimeno-Gómez, Catarina Botelho, Anna Pompili, Alberto Abad, and Carlos-D Martínez-Hinarejos. 2025. Unveiling interpretability in self-supervised speech representations for parkinson's diagnosis. *IEEE Journal of Selected Topics in Signal Processing*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press.

Kinan Martin, Jon Gauthier, Canaan Breiss, and Roger Levy. 2023. Probing self-supervised speech models for phonetic and phonemic information: a case study in aspiration. *arXiv preprint arXiv:2306.06232*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31.

Andrew Ng et al. 2011. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ankita Pasad, Chung-Ming Chien, Shane Settle, and Karen Livescu. 2024. What do self-supervised speech models know about words? *Transactions of the Association for Computational Linguistics*, 12:372–391.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. 2021. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 914–921. IEEE.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30.

Kenneth N Stevens. 2002. Toward a model for lexical access based on acoustic landmarks and distinctive features. *The Journal of the Acoustical Society of America*, 111(4):1872–1891.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

A HuBERT Parameters

The following section summarizes the parameters of the HuBERT model used for inference in our experimental setup.

Parameter	Value
feat_extract_activation	gelu
conv_bias	true
conv_dim	512
conv_kernel	[10, 3, 3, 3, 3, 2, 2]
conv_stride	[5, 2, 2, 2, 2, 2, 2]
attention_dropout	0.1
ctc_loss_reduction	sum
ctc_zero_infinity	false
feat_proj_dropout	0.1
final_dropout	0.1
hidden_dropout	0.1
hidden_act	gelu
hidden_dropout_prob	0.1
hidden_size	1024
intermediate_size	4096
layer_norm_eps	1e-5
layerdrop	0.1
mask_feature_length	10
mask_time_length	10
mask_time_prob	0.05
model_type	hubert
num_attention_heads	16
num_conv_pos_embedding_groups	16
num_conv_pos_embeddings	128
num_feat_extract_layers	7
num_hidden_layers	24
vocab_size	32

Table 2: Hyperparameter configuration of the HuBERT model used during experimentation. This information is available on Hugging Face.

B Data Splits

The following table shows the size of the TIMIT splits used during inference on HuBERT. For each raw waveform, we extract the HuBERT's last MLP activations.

Split	Audio files
Train	$\approx$ 4,620
Test	$\approx$ 1,680

Table 3: Splits of the TIMIT dataset.