AudioSAE: Towards Understanding of Audio-Processing Models with Sparse AutoEncoders

Georgii Aparin*, **Tasnima Sadekova***, **Alexey Rukhovich**, **Assel Yermekova**,
Laida Kushnareva, **Vadim Popov**, **Kristian Kuznetsov**, **Irina Piontkovskaya**
Huawei Noah's Ark Lab
aparingm@gmail.com, sadekova.t.r@gmail.com

Abstract

Sparse Autoencoders (SAEs) are powerful tools for interpreting neural representations, yet their use in audio remains underexplored. We train SAEs across all encoder layers of Whisper and HuBERT, provide an extensive evaluation of their stability, interpretability, and show their practical utility. Over 50% of the features remain consistent across random seeds, and reconstruction quality is preserved. SAE features capture general acoustic and semantic information as well as specific events, including environmental noises and paralinguistic sounds (e.g. laughter, whispering) and disentangle them effectively, requiring removal of only 19–27% of features to erase a concept. Feature steering reduces Whisper's false speech detections by 70% with negligible WER increase, demonstrating real-world applicability. Finally, we find SAE features correlated with human EEG activity during speech perception, indicating alignment with human neural processing. The code and checkpoints are available at https://github.com/audiosae/audiosae_demo.

1 Introduction

Audio and speech representation learning has evolved significantly over the past decades. Early models relied on mel-spectrograms, while recent advances in deep learning, particularly Transformers, have enabled large-scale audio modeling for speech recognition (Radford et al., 2022), synthesis (Borsos et al., 2023; Wang et al., 2023), and general audio understanding (Ghosh et al., 2025).

A major milestone has been the development of self-supervised learning (SSL) models such as Wav2Vec 2.0, WavLM, and HuBERT (Schneider et al., 2019; Baevski et al., 2020; Chen et al., 2022; Hsu et al., 2021), which learn directly from raw audio by predicting masked segments. Their intermediate representations have proven useful for diverse downstream tasks including automatic speech recognition (ASR), speaker verification, translation, and audio generation (Lee et al., 2021a,b; Van Niekerk et al., 2022; Kreuk et al., 2023).

In parallel, neural audio codecs (Zeghidour et al., 2021; Défossez et al., 2022; Zhang et al., 2023) and encoder part of models like Whisper (Radford et al., 2022) have emerged as universal feature extractors for speech LLMs. Trained on multilingual and noisy data, these encoders produce robust, semantically rich embeddings applicable across a wide range of audio tasks.

Since these models are optimized for different objectives, they encode information in distinct ways, motivating the need for systematic interpretability methods. Recently, *SAEs* have gained attention as a means to decompose dense representations into sparse, interpretable components (Cunningham et al., 2023a; Lieberum et al., 2024; Lim et al., 2025). While SAEs have been extensively studied in text and vision, their potential for analyzing audio models remains largely unexplored, with only isolated attempts in music modeling (Singh et al., 2025) or informal Whisper analyses.

Recent studies demonstrate that SAEs have been successfully applied across a wide range of models: from Gemma-2 (Lieberum et al., 2024) to GPT-4o (Wang et al., 2025), revealing interpretable internal mechanisms and enabling targeted behavior control. In NLP, SAEs help disentangle fine-grained semantic and stylistic features (Cunningham et al., 2023a), while Muhamed et al. (2025) used them to separate gender markers from professional content, mitigating spurious correlations. In computer vision, Cywiński and Deja (2025) achieved selective concept unlearning in diffusion models without additional fine-tuning. These results confirm that SAE features serve both as interpretable units of representation and as actionable levers for model steering.

In this work, we fill this gap by applying SAEs to large-scale audio representation models, Whisper

[*] Equal contribution

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3221–3254
March 24-29, 2026 ©2026 Association for Computational Linguistics

and HuBERT, and conducting a systematic evaluation of their stability, interpretability, and practical usefulness.

Our main contributions are as follows:

- We train SAEs on activations of Whisper and HuBERT and release the models and code, providing the first large-scale analysis of audio representation interpretability via sparse autoencoders.
- We develop distributional and automatic validation methods showing that SAE features are stable across seeds and encode semantic, paralinguistic, and acoustic information.
- We demonstrate practical and neuroscientific relevance by steering Whisper to reduce hallucinations and revealing correlations between SAE features and human EEG activity.

2 Related Works

2.1 Audio Representations

In this work we consider two popular models widely used in downstream tasks related to speech processing: Whisper (Radford et al., 2022) and HuBERT (Hsu et al., 2021). The former is an encoder-decoder Transformer trained with multi-tasking on a large corpus of $680k$ hours of multi-lingual speech, while the latter is encoder-only Transformer trained on $60k$ hours of English speech iteratively to predict labels assigned by the model in the previous iteration given masked audio sequence. We also experimented with EnCodec (Défossez et al., 2022) representations but omit them from the paper because the obtained latent space was not sufficiently sparse.

2.2 SAE in Various Domains and Applications

Natural Language Processing. Since (Sharkey et al., 2023) introduced the idea, Sparse Autoencoders (SAEs) have been extensively studied in the analysis of NLP models, with pretrained variants now available for GPT-2, Pythia (Cunningham et al., 2023a), Gemma-2 (Lieberum et al., 2024), LLaMA-3.1 (He et al., 2024), and other LLMs.

Spurious correlations detection in NLP. Muhamed et al. (2025) trained SAEs that separated gender markers from professional content in career biographies, improving classifier accuracy by eliminating gender-based shortcuts. Bricken et al.[1] trained a linear classifier on SAE features for bioweapon-related prompt classification and discovered an "academic formatting" feature that spuriously predicted harmlessness. Ablating it degraded classifier performance, and exploiting it allowed harmful prompts to be misclassified as harmless demonstrating SAE features' utility in identifying and eliminating spurious correlations.

SAE utility in downstream NLP tasks. Results on SAE utility in downstream tasks are mixed. Kantamneni et al. (2025) report that SAE features provide no advantage over linear probes for classification in several tasks. In contrast, other works demonstrate benefits of SAE features: Yang et al. (2025) outperform heuristics in data selection, Kuznetsov et al. (2025) achieve strong results in AI-generated text detection, and O'Neill et al. (2024) use SAE features for interpretable query steering in arXiv abstracts. Recently, Wang et al. (2025) successfully applied SAE to explain an important "emerging misalignment" phenomenon in LLM fine-tuning.

Computer Vision. SAEs enhance CV model interpretability (Kim et al., 2025) and image generation control (Stevens et al., 2025). In particular, Tian et al. (2025) and Cywiński and Deja (2025) achieve selective concept unlearning in diffusion models without fine-tuning.

Other domains. Simon and Zou (2024) and Gujral et al. (2025) apply SAEs to protein language models, generating interpretable features aligned with biological annotations.

In the **audio domain**, Singh et al. (2025) represents the only prior work, focusing on discovering concepts in music samples. We address this gap by training SAEs on more audio models and providing a substantially more comprehensive analysis of the learned features.

2.3 Evaluating SAE Quality

Karvonen et al. (2025) propose a comprehensive evaluation framework for SAEs in NLP, spanning eight metrics across four capabilities: Concept Detection (Feature absorption (Chanin et al., 2025), Sparse probing), Interpretability, Reconstruction Quality, and Feature Disentanglement (Unlearning, RAVEL (Huang et al., 2024), Targeted Probe Perturbation, Spurious Correlations removal). Testing over 200 SAEs reveals that improvements on sparsity-fidelity trade-offs don't always translate to better performance on practical tasks like unlearning or bias mitigation. The framework highlights that assessing multiple aspects of SAE performance

[1] https://transformer-circuits.pub/2024/features-as-classifiers/index.html

is improtant to determine which architecture is best suited for a given application, as SAEs show distinct task-specific strengths.

2.4 Other Interpretability Methods in Audio Domain

Wu et al. (2024) study neuron-level interpretability in acoustic models on environmental sound and music benchmarks. Their framework automatically generates natural-language explanations for acoustic neurons and leverages these explanations to cluster different sound types and to unlearn specific acoustic concepts. This work also sheds light on the kinds of features acoustic models rely on for sound classification and on how training strategies influence neuron interpretability.

Our work and Wu et al. (2024) are complementary in terms of both the models considered and the types of features analyzed. First, rather than examining embedding spaces directly, we study SAE-disentangled representations of these embeddings. Second, we extend our analysis to speech-related models and datasets. As a result, we identify a number of new feature types and clusters that are not discussed in (Wu et al., 2024).

3 Background and Methodology

3.1 SAE

Recent studies have shown that in transformer models, particularly in LLMs, individual neurons are *polysemantic* (Olah et al., 2020), meaning that these models learn more semantic features than there are available dimensions in a layer. This phenomenon is often explained by the *superposition hypothesis*, which suggests that polysemantic features are linear combinations of monosemantic ones, allowing models to represent more features than they have dimensions (Elhage et al., 2022; Hänni et al., 2024).

To recover these features, a Sparse Autoencoder (SAE) learns directions in the activation space such that each activation is a sparse linear combination of them. Given activations $\boldsymbol{x}$, the SAE encodes and reconstructs them as

$$\begin{aligned} f(\boldsymbol{x}) &= \sigma(\boldsymbol{W}_{\text{enc}}\boldsymbol{x} + \boldsymbol{b}_{\text{enc}}), \\ \hat{\boldsymbol{x}}(f(\boldsymbol{x})) &= \boldsymbol{W}_{\text{dec}} f(\boldsymbol{x}) + \boldsymbol{b}_{\text{dec}}, \end{aligned} \tag{1}$$

where σ is an activation function such as ReLU, Top-k or Batch-Top-k (Cunningham et al., 2023b; Gao et al., 2025; Bussmann et al., 2024).

The SAE is typically optimized using the reconstruction loss $\mathcal{L}_{\text{rec}}(\boldsymbol{x}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}(f(\boldsymbol{x}))\|_2^2$, with an additional sparsity loss $\mathcal{L}_{\text{sp}}(\boldsymbol{x})$ or auxiliary loss $\mathcal{L}_{\text{aux}}(\boldsymbol{x})$, depending on the architecture.

In multiple previous works, listed in the preceding section, it was demonstrated that SAEs are indeed capable of extracting monosemantic features. A theoretical justification for this phenomenon was later provided in (Cui et al., 2025). Most prior applications have focused on NLP and the visual domain, while audio SAEs remain comparatively less explored.

3.2 Audio SAE Training Setup

The SAE has a relatively straightforward architecture, with the main design choice being the form of the non-linearity σ, which influences the sparsity patterns learned by the model. Among the three explored options, Jump-ReLU, Top-k, and Batch-Top-k, we found the Batch-Top-k variant to perform slightly better in terms of both reconstruction quality and sparsity. Therefore, it was selected for all subsequent experiments. All SAEs were trained using an $\mathcal{L}_2$ reconstruction loss without any auxiliary regularization.

3.3 SAE Evaluation and Analysis

Evaluating SAEs is challenging, as no single metric fully captures their quality. Following Karvonen et al. (2025), we adopt a multi-faceted evaluation covering reconstruction, robustness, interpretability, and disentanglement. We report $\mathcal{L}_2$–$\mathcal{L}_0$ trade-offs for reconstruction quality, assess feature stability across random seeds, layers, and models, and apply a set of interpretability methods to analyze the latent space.

We conduct both frame-level analysis and audio-level analysis using max-pooled feature aggregation.

3.3.1 Feature Robustness via Distributional Semantics

To measure feature stability, we introduce a distributional similarity metric inspired by distributional semantics (Harris, 1954; Mikolov et al., 2013). Two features a_k, b_m are considered *semantically similar* if their binary activation patterns over dataset $\mathcal{D} = \{d_i\}$ have high Intersection-over-Union (IoU):

$$\chi(a_k, b_m) = \frac{|\{i \mid a_k(d_i) = 1 \wedge b_m(d_i) = 1\}|}{|\{i \mid a_k(d_i) = 1 \vee b_m(d_i) = 1\}|}$$

A feature a_k is *covered* by set B if $\exists b_m \in B : \chi(a_k, b_m) > \theta$, and *coverage*

$$c(A, B) = |\{k \mid \exists m : \chi(a_k, b_m) > \theta\}|$$

quantifies the proportion of transferable features. We apply this measure to compare sets of features of SAEs for different random seeds, layers, and model architectures (HuBERT, Whisper). The high coverage would indicate presence of robust and consistent features. Also we define the *duplicated features* of some SAE as those having high IoU with some other feature within the same SAE. The high amount of duplicates would indicate redundancy among SAE features.

3.3.2 Domain Specialization

We analyze high-level organization by attributing features to one of three domains: *speech*, *music*, and *environmental sounds*. A feature is considered *domain-specialized* if its activation frequency is significantly higher within domain D than in its complement $\overline{D}$. Activation frequency is assessed at two distinct levels for each domain D. At the frame level, it is defined as the proportion of frames within all samples in the domain that exhibit non-zero activation for a given feature. At the audio level, it is defined as the proportion of audios in the domain in which the feature is activated at least once.

3.3.3 Interpretability Analysis

To validate the general applicability of SAE features for audio-level analysis, we assess their performance across several **classification** tasks. These include gender identification, noise condition classification (clean vs. noisy speech), as well as accent and emotion recognition. All latents are ranked according to their influence on classification performance using the Fisher score (Fisher, 1936). To examine the impact of the highest-ranked features, we employ two techniques: *top-k probing*, which masks all features except the top-k during activations' reconstruction, and *unlearning*, which involves masking the top-k features (Karvonen et al., 2025).

Additionally, fine-grained interpretability is assessed through the following methods:

- Manual inspection of **top-activated samples** from *reference set* of the most representative intervals. It consists of 1000 samples from LibriTTS, Expresso, ESD, FSD50k and ESC50 datasets;
- **Semantic analysis**. To assess whether the SAE latent space captures speech semantics, we perform two experiments: vowel pronunciation classification and phoneme alignment. The first task addresses the complex phenomenon of how characters are pronounced, while the second task enables the mapping of some SAE features to specific linguistic units, namely phonemes;
- **Label-based search.** We identify features that demonstrate a strong individual correlation with specific label from the reference set (e.g., speech/non-speech, emotion, sound type) and lead to a high accuracy in separating data with target label from all others;
- **Mel-interpretation** via averaging mel-spectrograms representation of activated frames to identify recurrent acoustic patterns;
- **Auto-interpretation via captioning.** Active frames for a concrete feature from top-activated samples are concatenated to form 2-second chunks, which are processed by an audio captioning model. A large language model then aggregates all generated captions to produce a unified, high-level description of the feature;

Together, these analyses provide a comprehensive view of SAE robustness, interpretability, and functional relevance.

3.4 Hallucination Reduction via SAE Steering

Steering refers to linear interventions in the latent space that guide model behavior toward desired outcomes. In the audio domain, we apply this approach to mitigate *hallucinations* in the Whisper model (Barański et al., 2025), where hallucination denotes false speech predictions in non-speech segments. To reduce such errors, we add a directional vector during inference that biases activations away from hallucination-prone regions.

As a proxy metric for assessing the Whisper's tendency to hallucinate, we use an internal parameter no_speech_prob – the probability that there is no speech in the audio. We aim to shift the distribution of this parameter toward 1 for non-speech samples and toward 0 for speech samples.

We search for SAE features that characterize hallucinations and modify them in the SAE space to minimize hallucinations. We define the Detection

Rate (DR) metric as:

$$\mathrm{DR}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} \mathbb{I}\{\text{no\_speech\_prob}_i < \tau\}.$$

This metric quantifies the fraction of samples classified as speech-containing. For non-speech datasets, DR estimates the False Positive Rate (FPR), capturing the proportion of hallucinations (non-speech samples incorrectly classified as speech). For speech datasets, DR estimates the True Positive Rate (TPR), the proportion of correctly identified speech samples.

To identify the top-k SAE features related to hallucinations, we train a logistic regression classifier on SAE activations for non-speech dataset on target of predicting hallucinations (no_speech_prob $<$ τ). SAE feature j is included in top-k features if the absolute value of its regression coefficient β_j is among largest k such values. Then the SAE steering vector is defined as:

$$\vec{s}_{\text{SAE}}[j] = -\text{sign}(\beta_j), \quad j \in \text{top-}k,$$

and zero elsewhere. The negation operator ($-\text{sign}(\beta_j)$) ensures that the steering vector is opposing to features positively associated with hallucinations, thereby steering the model away from hallucination-prone representations.

During inference activations modified as

$$\text{act}_{\text{steered}} = \hat{\boldsymbol{x}}(f(\text{act}) + \alpha\vec{s}_{\text{SAE}}),$$

where α controls steering intensity, f and $\hat{\boldsymbol{x}}$ are SAE's Encoder and Decoder respectively, see Formula 1.

3.5 Correlation with EEG

We closely follow the experimental setting from (Broderick et al., 2018) and use freely available data provided in this paper. Its authors tried to find correlation between semantic dissimilarity stimuli s and EEG response r by fitting a linear model representing response as a convolution of stimuli with some filter expressed in time domain by the so-called temporal-response function (TRF) w:

$$r(t) = \sum_{\tau} w(\tau)s(t-\tau) + \varepsilon(t), \qquad (2)$$

where ε is a residual signal whose energy is minimized during model fitting (i.e. finding TRF). If $w(\tau)$ differs from zero significantly for some τ, then we can deduce that stimuli correlate with EEG response signal with time lag τ. Note that, in contrast with our paper, Broderick et al. (2018) used the term "semantic" in pure NLP sense corresponding to meanings of particular words – they studied semantic dissimilarity stimuli s measuring how unexpected a word is given its left context by computing word2vec embeddings (Mikolov et al., 2013). In this paper we treat SAE features as stimuli without prior assumptions on what exactly they represent. We compute TRFs and do statistical tests to find out whether some of these features represent concepts aligned with EEG activity of people listening to speech.

4 Experiments

4.1 Training Details

Base models. We trained SAEs on the HuBERT and Whisper model families. For our downstream analysis, we focus on the HuBERT-base[2] and Whisper-small[3] variants. Activations were extracted from every layer of the models' encoders.

Data. The models were trained on a diverse corpus of approximately 2.8k hours of audio, encompassing speech, music, and environmental sounds. A complete list of datasets included in the corpus is provided in Table 3 (Appendix A).

To improve robustness, we apply online augmentation by additively mixing noise and music at probabilities $p_{noise} = 0.05$ and $p_{music} = 0.025$, with signal-to-noise ratios uniformly sampled between 0 – 20 dB. Activation vectors are stored in memory-mapped buffers and shuffled to enable efficient, randomized sampling.

Hyperparameters. All SAEs used a BatchTopK architecture and were trained with the Adam optimizer for 200,000 steps using an $\mathcal{L}_2$ reconstruction loss. More details in Appendix A.

4.2 SAE Quality Evaluation

Reconstruction–sparsity trade-off. Fig. 1 shows the trade-off between SAE sparsity and reconstruction quality under different training hyperparameters. Varying k (the number of active neurons per token) yields Pareto frontiers at fixed expansion rates. Both expansion factors demonstrate comparable results in term of reconstruction quality. We

[2] https://huggingface.co/facebook/hubert-base-ls960

[3] https://huggingface.co/openai/whisper-small

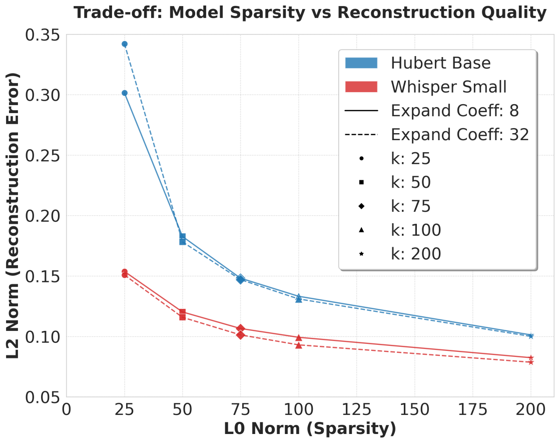


Figure 1: The trade-off between normalized reconstruction error (L2) and sparsity (L0) for models at layer 12.

Model	**L1**	**L4**	**L7**	**L10**	**L12**
Hub_Hub^2	419	1768	2930	4295	3164
Hub_Hub$_{L+n}$	92	258	453	1466	
Wh_Wh$_{L+n}$	921	1296	634	2692	
Hub_Wh	50	95	65	125	180
Hub (dup)	27	47	102	352	219
Wh (dup)	793	755	787	221	230
Gem_Gem$_{L+n}$	$^{\mathbf{L2}}$3365	$^{\mathbf{L6}}$3239	$^{\mathbf{L12}}$3210	$^{\mathbf{L18}}$2901	
Gem (dup)	$^{\mathbf{L2}}$4741	$^{\mathbf{L6}}$7825	$^{\mathbf{L12}}$3174	$^{\mathbf{L18}}$7115	$^{\mathbf{L24}}$2734

Table 1: SAE feature set coverage between models and layers. Suffix ^{2} is for SAE trained on the same activations but initialized with different random seeds; suffix $_{L+n}$ is for coverage between different layers of the model (each layer is compared with the layer from the next column, i.e. in L4 column we show the coverage between features from layer 4 with features from layer 7); (dup) means the amount of duplicates. For Gemma model we took SAEs from layers 2, 6, 12, 18 and 24. Note that the number of features for our SAEs in 6144, and for Gemma-Scope is 16384.

therefore adopt SAEs with $k = 50$ and 8x expansion for all subsequent experiments. Additional experiments are provided in Appendix A.

Feature Robustness Evaluation. We evaluate feature stability using the distributional metrics described in Section 3.3.1. Table 1 reports results for several layers of HuBERT and Whisper on LibriSpeech; additional layers and datasets are shown.

Amount of duplicates within a single checkpoint are low: below 5% for the final layers of both models. In contrast, feature coverage between checkpoints trained on the same layer but with different random seeds exceeds 50%, indicating strong stability. However, features learned by HuBERT do not align with those from Whisper, likely due to differences in pre-training data and objectives: HuBERT is trained via self-supervision on ASR data, whereas Whisper is trained through supervised learning for a broad range of audio tasks. The inter-layer feature coverage is high only for latter layers.

For comparison, we measure the duplicate count and inter-layer coverage for GemmaScope SAEs (Lieberum et al., 2024), and show that our SAEs exhibit lower redundancy and comparable inter-layer coverage at last layers. Overall, our SAEs extract a stable set of conceptual features from activations, although these concepts differ across models. Additional results are provided in Appendix B.

4.3 Domain Specialization Analysis

Our domain analysis of feature type distributions across all 12 layers of both Whisper and HuBERT revealed distinct layer specialization trends (Fig. 2).

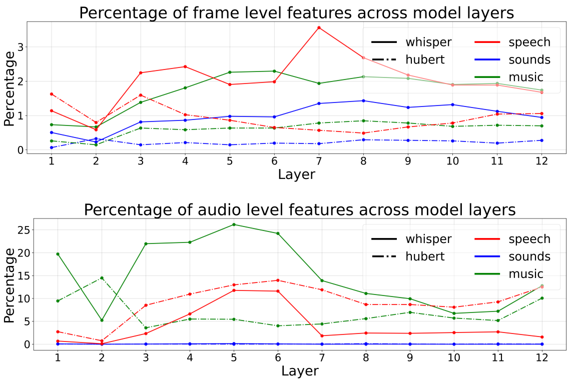


Figure 2: Layer-wise feature specialization ratio by speech, sounds, and music domains for Whisper (solid line) and HuBERT (dashed) at frame (top) and audio (bottom) levels.

Whisper exhibits pronounced audio-level specialization for music, with music features comprising roughly 20–28% of the detected audio-level activations and peaking around layer 5 (Fig. 2, bottom). Speech-related audio-level features are also most prominent in mid layers (peaking at roughly 13%), but decrease sharply after layer 6, reaching only a few percent by layer 7.

At the frame level, Whisper's speech specialization peaks later: the speech-features proportion rises from about 2% at layer 6 to approximately 3.5% at layer 7 (Fig. 2, top), with the concurrent drop in audio-level speech activations. This divergence suggests that some layers encode speech information more locally (frame-level) even when global (audio-level) features are less frequently activated. Further analysis and additional experiments are described in Appendix C.5 and Appendix C.2 respectively.

4.4 Classification-based Analysis

First, we analyze learned features via classification (Section 3.3.3) on four audio tasks: gender (2 classes), clean vs. noisy speech (2), accents (5), and emotions (5). Logistic regression is chosen as the classifier, with a one-vs-all strategy applied for the 5-class tasks. First, a classifier is trained on SAE features to rank their importance for a specific task. To assess the influence of these features on model activations, classifiers are subsequently trained on the reconstructed activations. Results on accents are shown in Fig. 3; additional details are in Appendix D.

The full SAE does not degrade performance and may even improve it (e.g., on emotion tasks). The top-k curves rise sharply and saturate quickly, indicating that a small number of features ($k \approx 10$–150 out of 6144 for binary tasks and 500–3000 for more complex multi-class objective) captures most task-relevant information.

However, removing this information completely requires suppressing many more features ($\sim$2000), showing redundancy and distributed encoding: complex traits such as accent depend on multiple cues (phonemes, prosody, intonation). Compared to random selection, both top-k and unlearning curves converge faster, confirming that learned features encode meaningful structure.

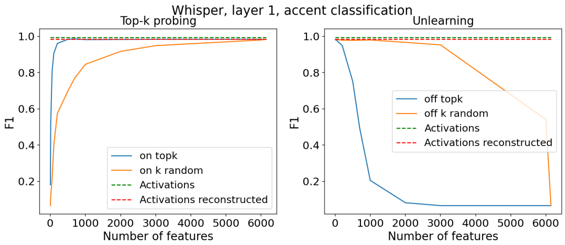


Figure 3: Top-k probing and unlearning for accent classification. More results in Appendix D.

4.5 Semantic Analysis

Letter pronunciation classification. To demonstrate *disentanglement*, we employ a vowel unlearning experiment: if disentangled features exist for different phonemes, we should be able to selectively unlearn one phoneme class while preserving recognition of others. Using AVLetters2 (Cox et al., 2008) – recordings of five speakers pronouncing English letters – we focus on vowels due to their simpler, atomic articulations. Fig. 4 shows sequential removal of features most discriminative for letter "A". Deleting the first 1152 features ($\approx$19%) almost entirely erases "A", while recognition of other vowels (MCC (Matthews Correlation Coefficient) > 0.75) remains stable until over 27% of features are removed – showing some *disentangled* features. See Appendix E for more results.

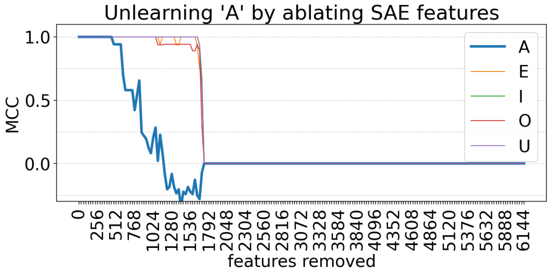


Figure 4: Selective unlearning of letter 'A' via iterative feature removal. Feature indices on x-axis ordered by discriminative importance for target vowel.

Overall, erasing speech concepts requires far more features (hundreds to thousands) than text-based SAEs, where abstract notions like gender or occupation can be removed with only tens (Farrell et al., 2024). This reflects both (a) higher redundancy and (b) the inherently distributed nature of phonetic and paralinguistic information.

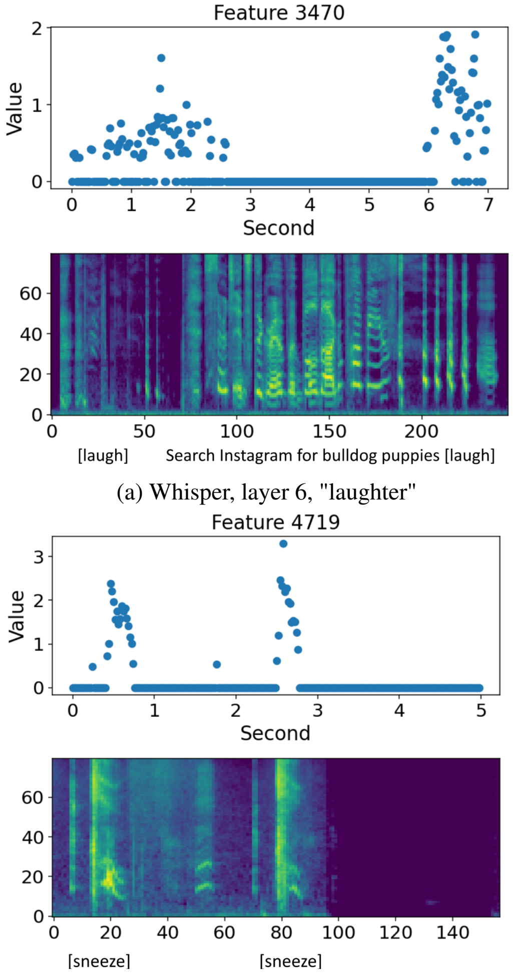
Phonemes encoding. To additionally verify which SAE features encode semantic information we work with text-audio alignments[4] extracted from 1000 audio samples from LibriTTS. A phoneme label is assigned to a latent feature when this phoneme appears in a majority ($> 50\%$) of its aligned, activated frames.

The final 12th layers of the Whisper and HuBERT models are analyzed on a test set consisting of 1000 audio samples from LibriTTS. A frame is considered correctly classified if at least one activated SAE feature (determined by a threshold) has a label matching the ground-truth phoneme. The final accuracy scores achieved for these models are 0.92 and 0.89, respectively.

4.6 Frame-level Features Interpretation

By **label-based classification**, we aim to identify individual SAE features that independently encode concepts present in the dataset and separate label-

[4] Pre-trained aligner from `https://montreal-forced-aligner.readthedocs.io/`



(a) Whisper, layer 6, "laughter"

(b) Whisper, layer 6, "sneezing"

Figure 5: Features identified via the label-based classification experiment. Each sub-figure displays a feature's values above the corresponding audio mel-spectrogram.

specific samples from all others by some threshold. The label *whisper* was successfully detected by HuBERT SAEs on layers 1-5, with feature 6106 (layer 4) achieving an F1-score of 0.6. Similarly, *laughter* is captured by both models – on layers 1–6 in HuBERT and layers 1, 6, and 9 in Whisper. The phenomena *sigh* and *sneezing* are also represented, with relevant features appearing across layers 1-7 in both models. In contrast, *animal sounds* and *breathing* were not identified reliably, suggesting that these concepts are encoded in a distributed manner. Frame-level examples are shown in Fig. 5 and Appendix F, with an extended version available on the demo page.

Using the **mel-interpretation methodology** we further analyze salient features identified through domain specialization. Specifically, we extract 1-second log-mel spectrogram windows centered at activated frame from the audio samples with highest activation magnitude. The element-wise average of these windows reveals the core acoustic pattern. Thus we found that HuBERT layer 11 features 3249 and 3081 exhibit specialized speech-boundary detection. Their temporal alignment with speech segments is illustrated in Appendix I.

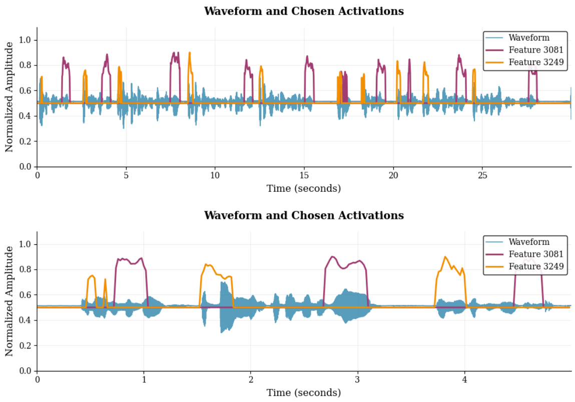


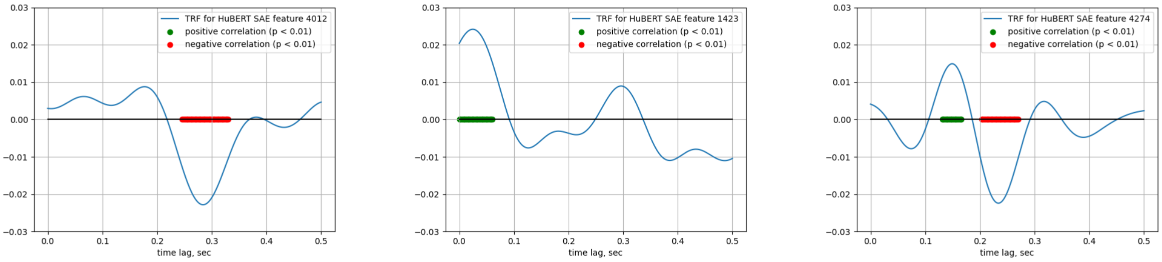
Figure 6: Activation of features responsible for the beginning (3249) and the end (3081) of speech, aligned with corresponding waveform. HuBERT, layer 11.

For the **automatic interpretation** of features, we first extract all time frames in the audio where a feature's value exceeds the threshold of 0.1, marking them as activated frames. These frames are concatenated and divided into 2-second segments, each processed by captioning model (Xu et al., 2024). The resulting captions are then unified and aggregated using model GPT-4o mini (OpenAI et al., 2024) to produce a final interpretative label for the feature. The entire pipeline is visualized in Appendix G, Fig. 15.

Auto interpretation helped identify unique features not present in the dataset annotations, such as **"ringing alarms"**, **"high-pitched beeping"**, **"birds chirping"**, and **"guitar playing"**. However, due to the limitations of the caption model, which was trained mainly on music and sound data, specific speech features, particularly, phonetic details were missed. As an example, the feature, which activates on the **"ba"** sound, resulted in a more general interpretation **"a man is speaking."** More detailed observation of the results can be seen in the Appendix G.

4.7 Steering for Hallucination Reduction

To evaluate the effectiveness of SAE in reducing hallucinations (Sec. 3.4), we measure the False Positive Rate (FPR) with $\tau = 0.5$ on non-speech datasets (FSD50k, Musan and WHAM) and control Word Error Rate (WER) on LibriSpeech test-clean



(a) only negative correlation

(b) only positive correlation

(c) both positive and negative correlation

Figure 7: Temporal response functions for different SAE features for HuBERT model.

to ensure that speech recognition performance remains unaffected.

Optimal performance is achieved by steering the top-100 SAE features. We report SAE steering with different strengths ($\alpha = 1$ and $\alpha = 3$) and compare it with the baseline S-vector steering approach (described in Appendix H) with $\alpha = 3$. Both methods substantially reduce hallucinations. SAE-based steering with moderate α provides a good balance between FPR and WER, achieving a threefold reduction in FPR by 70% ($0.37 \rightarrow 0.11$) in average across datasets with only a negligible increase in WER ($5.1\% \rightarrow 5.5\%$). However, aggressive steering with large α severely impairs speech comprehension, revealing a clear trade-off between efficacy and safety. Table 2 summarizes the results. More details in Appendix H.

Dataset	**No SAE**	**No Steer.**	**S-Vec.** (α=3)	**SAE** (α=1)	**SAE** (α=3)
Musan	-	0.33	0.20	0.12	0.00
FSD50K	-	0.26	0.15	0.09	0.01
WHAM	-	0.51	0.32	0.14	0.03
LS (WER)	5.1	5.2	5.3	5.5	98.4

Table 2: FPR ($\tau = 0.5$) for steering configurations. No SAE: Whisper inference without modification. No Steer.: Whisper with injected SAE on the last layer. S-Vec.: S-Vector, calculated on Musan. Optimal and Best SAE: SAE S-Vector, top-100 features from FSD50k dataset with α equals to 1 and 3 respectively. LibriSpeech (LS) line represents WER. Lower is better.

4.8 Correlation with EEG

Here we present EEG signal correlation experiments (Sec. 3.5). Open-source EEG data was collected from 19 participants listening to audiobooks. We studied both HuBERT and Whisper SAE features and found that some of them have statistically significant correlation with midline parietal electrode Pz (chosen as one of the most indicative in (Broderick et al., 2018)) at certain time lags τ as verified by one-tailed t-tests (p-value less than 0.05) with Holm-Bonferroni correction for multiple comparisons. As illustrated in Fig. 7, this correlation can be both positive and negative and occur at quite different time lags between 0 and 500ms. We further analyzed these features and found that many of them activate mostly on particular vowels (like IPA phonemes “ɔ” or “ɑ”), but not on all of such vowels. This analysis, more details of which can be found in Appendix J, shows that at least some of features learned by SAE are generic features well-aligned with brain activity. Interpreting them and analyzing their TRF patterns, as well as studying EEG channels other than Pz and applying more sophisticated non-linear models rather than (2) can be a future research direction.

5 Conclusion

This work presents a comprehensive investigation into the application of SAEs for interpreting the HuBERT and Whisper audio models. We introduce a novel metric for cross-layer and cross-model SAE evaluation and a bunch of methods for resulting latent analysis. Our proposed metric confirms that the resulting SAE features are robust and encode meaningful information. We found high-level features related to broad categories like speech and music, as well as more fine-grained features corresponding to semantic content (phonemes), paralinguistic phenomena (e.g., laughter, sigh, sneezing), and acoustic properties (discovered via autointerpretation). Furthermore, steering on Whisper model SAE features reduced the false positive rate on hallucinations by 70%. An additional experiment demonstrated correlation between EEG signals and specific SAE features.

Limitations

- Our downstream evaluation covers a limited set of classification tasks and applications. Future work should explore SAE features across broader audio processing tasks including speaker verification, speech enhancement, and audio generation.

- Detailed analysis focuses on base/small model variants. Larger architectures and additional models (Wav2Vec 2.0, WavLM) were not comprehensively studied due to computational constraints.

- The auto-interpretation method inherits limitations from its underlying audio captioning model, which was trained primarily on music and sound data. As a result, it tends to generate generic captions for speech-related features, losing fine-grained phoneme-level information.

- EEG correlation analysis is limited to a single electrode (Pz) with linear temporal response models. More comprehensive brain imaging and non-linear modeling could reveal additional relationships.

References

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

Mateusz Barański, Jan Jasiński, Julitta Bartolewska, Stanisław Kacprzak, Marcin Witkowski, and Konrad Kowalczyk. 2025. Investigation of whisper asr hallucinations induced by non-speech audio. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 1–5. IEEE.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. 2019. The mtg-jamendo dataset for automatic music tagging. ICML.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533.

Cassia Valentini Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. 2016. Investigating rnn-based speech enhancement methods for noise-robust text-to-speech. In *9th ISCA speech synthesis workshop*, pages 159–165.

Michael P. Broderick, Andrew J. Anderson, Giovanni M. Di Liberto, Michael J. Crosse, and Edmund C. Lalor. 2018. Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech. *Current Biology*, 28(5):803–809.e3.

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

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. 2025. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *Preprint*, arXiv:2409.14507.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 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.

Stephen Cox, Richard Harvey, and Yuxuan Lan. 2008. The challenge of multispeaker lip-reading.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. 2025. On the theoretical understanding of identifiable sparse autoencoders and beyond. *arXiv preprint arXiv:2506.15963*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023a. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023b. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Bartosz Cywiński and Kamil Deja. 2025. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. 2024. Applying sparse autoencoders to unlearn knowledge in language models. *Preprint*, arXiv:2410.19278.

R. A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. 2021. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. 2025. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. *Preprint*, arXiv:2503.03983.

Yuan Gong, Jin Yu, and James Glass. 2022. Vocalsound: A dataset for improving human vocal sounds recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 151–155. IEEE.

Onkar Gujral, Mihir Bafna, Eric Alm, and Bonnie Berger. 2025. Sparse autoencoders uncover biologically interpretable features in protein language model representations. *Proceedings of the National Academy of Sciences*, 122(34):e2506316122.

Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. 2024. Mathematical models of computation in superposition. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

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.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024. RAVEL: Evaluating interpretability methods on disentangling language model representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8669–8687, Bangkok, Thailand. Association for Computational Linguistics.

Fanny Jourdan, Louis Béthune, Agustin Picard, Laurent Risser, and Nicholas Asher. 2024. Taco: Targeted concept erasure prevents non-linear classifiers from detecting protected attributes. *Preprint*, arXiv:2312.06499.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2024. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995. IEEE.

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

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *Preprint*, arXiv:2503.09532.

Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. 2025. *Revelio*: Interpreting and leveraging semantic information in diffusion models. *Preprint*, arXiv:2411.16725.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2023. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*.

Kristian Kuznetsov, Laida Kushnareva, Anton Razzhigaev, Polina Druzhinina, Anastasia Voznyuk, Irina Piontkovskaya, Evgeny Burnaev, and Serguei Barannikov. 2025. Feature-level insights into artificial text detection with sparse autoencoders. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25727–25748, Vienna, Austria. Association for Computational Linguistics.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Sravya Popuri, Xutai Ma, Adam Polyak, Yossi

Adi, Qing He, Yun Tang, and 1 others. 2021a. Direct speech-to-speech translation with discrete units. *arXiv preprint arXiv:2107.05604*.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, and 1 others. 2021b. Textless speech-to-speech translation on real data. *arXiv preprint arXiv:2112.08352*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. 2025. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *The Thirteenth International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Aashiq Muhamed, Mona T. Diab, and Virginia Smith. 2025. Decoding dark matter: Specialized sparse autoencoders for interpreting rare concepts in foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1604–1635, Albuquerque, New Mexico. Association for Computational Linguistics.

Tu Anh Nguyen, Wei-Ning Hsu, Antony d'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, and 1 others. 2023. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *arXiv preprint arXiv:2308.05725*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F. Wu. 2024. Disentangling dense embeddings with sparse autoencoders. *Preprint*, arXiv:2408.00657.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Karol J Piczak. 2015. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 527–536.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Muhammad Mamunur Rashid, Guiqing Li, and Chengrui Du. 2023. Nonspeech7k dataset: Classification and analysis of human non-speech sound. *IET Signal Processing*, 17(6):e12233.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 2022. Linear adversarial concept erasure. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18400–18421. PMLR.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Lee Sharkey, Dan Braun, and Beren Millidge. 2023. Taking features out of superposition with sparse autoencoders. *URL https://www.lesswrong.com/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition*.

Elana Simon and James Zou. 2024. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *Preprint*, arXiv:2412.12101.

Nikhil Singh, Manuel Cherep, and Pattie Maes. 2025. Discovering and steering interpretable concepts in large generative music models. *Preprint*, arXiv:2505.18186.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. 2025. Sparse autoencoders for scientifically rigorous interpretation of vision models. *Preprint*, arXiv:2502.06755.

Zhihua Tian, Sirun Nan, Ming Xu, Shengfang Zhai, Wenjie Qu, Jian Liu, Kui Ren, Ruoxi Jia, and Jiaheng Zhang. 2025. Sparse autoencoder as a zero-shot classifier for concept erasing in text-to-image diffusion models. *CoRR*, abs/2503.09446.

Benjamin Van Niekerk, Marc-André Carbonneau, Julian Zaïdi, Matthew Baas, Hugo Seuté, and Herman Kamper. 2022. A comparison of discrete and soft speech units for improved voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6562–6566. IEEE.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Miles Wang, Tom Dupré la Tour, Olivia Watkins, Alex Makelov, Ryan A Chi, Samuel Miserendino, Johannes Heidecke, Tejal Patwardhan, and Dan Mossing. 2025. Persona features control emergent misalignment. *arXiv preprint arXiv:2506.19823*.

Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. 2019. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*.

Tung-Yu Wu, Yu-Xiang Lin, and Tsui-Wei Weng. 2024. AND: Audio network dissection for interpreting deep acoustic models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53656–53680. PMLR.

Xuenan Xu, Haohe Liu, Mengyue Wu, Wenwu Wang, and Mark D. Plumbley. 2024. Efficient audio captioning with encoder-level knowledge distillation. *Preprint*, arXiv:2407.14329.

Xianjun Yang, Shaoliang Nie, Lijuan Liu, Suchin Gururangan, Ujjwal Karn, Rui Hou, Madian Khabsa, and Yuning Mao. 2025. Diversity-driven data selection for language model tuning through sparse autoencoder. *Preprint*, arXiv:2502.14050.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *Preprint*, arXiv:2107.03312.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. 2021. Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 920–924. IEEE.

Appendix

Table of Contents

A Extended SAE training details

This section provides additional details on model selection, dataset construction, and architectural choices referenced in the main text. During architecture search and hyperparameter sweeps, we evaluated trade-offs among the following metrics: reconstruction quality (L_2 loss, lower is better), sparsity (L_0 loss, lower is better), and the proportion of features activated at least once during N steps (“alive”; higher is better).

Base model Selection. Our study included SAE training on four model variants: HuBERT-base, HuBERT-large, Whisper-small, and Whisper-large-v3-turbo. While SAEs were trained for all variants to ensure a comprehensive foundation, the downstream analysis in the main paper is conducted on HuBERT-base and Whisper-small for a focused comparison. We also initially considered the EnCodec model. However, a SAE trained on the final layer of its encoder yielded a number of active ("alive") features comparable to the source embedding dimension, suggesting it was not learning a sufficiently sparse representation for our purposes, and it was excluded from further analysis.

The HuBERT, Whisper, and Montreal Forced Aligner (MFA) software packages are distributed under the MIT license.

Dataset. Our training corpus is designed to comprehensively represent diverse acoustic environments. The dataset is composed of multiple publicly available sources, with each assigned a sampling weight to control its prevalence during activation extraction and training. All datasets used are described in the Table 3.

The high weights for datasets like MUSAN, FSD50K, and Nonspeech7k were chosen to strongly bias the SAEs towards learning features for non-speech audio, music, and environmental sounds, complementing the speech-dominant datasets. After analyzing the audio types in the datasets, we decided to divide them into three types: *speech* (LibriSpeech (Panayotov et al., 2015), LibriHeavy (Kang et al., 2024), ESD (Zhou et al., 2021), Expresso (Nguyen et al., 2023), CREMA (Cao et al., 2014), MELD (Poria et al., 2019), IEMOCAP (Busso et al., 2008)), *music* (MTG-Jamendo (Bogdanov et al., 2019)), and *sounds* included noise, sound events and non-speech sounds (MUSAN (Snyder et al., 2015), WHAM (Wichern et al., 2019), FSD50K (Fonseca et al., 2021), Nonspeech7k (Rashid et al., 2023), DEMAND (Botinhao et al., 2016), VGGSound (Chen et al., 2020), VocalSound (Gong et al., 2022), ESC-50 (Piczak, 2015)). Taking into account the weights, each batch averaged approximately 40% activations from speech data, 45% from music, and 15% from sounds. With a batch size of 2500, a step count of 200,000, and a model frame rate of 50 frames per second, the total data amount for each SAE was just under 2800 hours.

Batching. Our training employs a dynamic batching strategy: audio samples are drawn from our weighted dataset mixture using a probability-proportional-to-size sampling scheme, where each dataset’s selection probability is determined by its configured weight multiplied by the number of sam-

Table 3: Composition of the Training Corpus

Dataset	Description	Weight	Hours
LibriSpeech	Read Speech	1.0	960
LibriHeavy	Noisy Speech	0.1	50000
MUSAN	Music/Speech/Noise	5	112
WHAM!	Domestic Noise	2	58
FSD50K	Sound Events	5	108
Nonspeech7k	Non-Speech Sounds	5	125
ESD	Emotional Speech	6	11
Expresso	Audiobooks	5	41
CREMA-D	Emotional Speech	4	5
DEMAND	Env. Noise	3	6
MELD	Emotional Dialogues	3	12
VGGSound	Audio Events	1	550
VocalSound	Vocalizations	3	18
MTG-Jamendo	Music	2	3777
IEMOCAP	Emotional Speech	Test only	12
ESC-50	Sound Classification	Test only	3

ples in each dataset. We randomly select a dataset determined by its weight and size, randomly select audio samples from a chosen dataset, then fulfill the buffer by model's activations on this audio sample and sample fixed-size batches by randomly selecting unread indices from the buffer, ensuring diverse training examples. This approach helps prevent overfitting to the sequence order of any single audio file.

Infrastructure. Training was conducted on a multi-GPU server. The inference of the base audio models (Whisper and HuBERT) was distributed across several GPUs to parallelize the computationally intensive forward passes required for activation generation. The subsequent SAE training was also sharded across available devices. SAEs were trained in parallel across all model layers within a single run using multi-threaded execution on identical data. We implemented an asynchronous data loading and buffering pipeline which pre-computes and stores activations in a memory buffer (holding 100 batches of 2500 activation vectors each), which is then sampled randomly to feed the SAE trainers. All experiments were performed on 8 NVIDIA V100 GPUs.

Training hyperparameters. We employ the Adam optimizer $(\beta_1, \beta_2) = (0.9, 0.999)$ with a fixed learning rate of 2×10^{-4} and a linear warmup of the sparsity coefficient over the first 10,000 steps. Training proceeds for 200,000 update steps, with a linear learning rate decay schedule initialized for the terminal 20% of training, progressively reducing the rate from its initial value to zero. Each training batch comprises 2,500 activation vectors, corresponding to 50 seconds of audio.

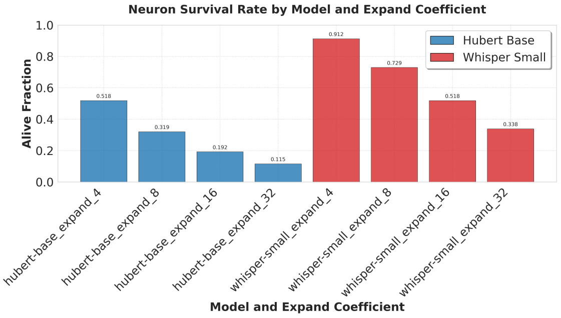


Figure 8: Influence of the expansion rate to the number of "alive" features

SAE architecture Selection. We tested three different SAE variants, depending of the form of non-linearity function: the Jump-ReLU, Top-K, and Batch-Top-K. Our preliminary analysis indicated that the Batch-Top-K SAEs demonstrated a slightly better performance in terms of reconstruction quality and sparsity control. Consequently, the Batch-Top-K was selected as the primary architecture for our investigation. All SAEs were optimized using an $\mathcal{L}_2$ reconstruction objective, without any auxiliary regularization.

SAE-specific hyperparameters. Key SAE hyperparameters include the *expansion factor* (ratio of SAE to source embedding dimensions) and the *sparsity level* k (number of active features per training sample). We perform a structured sweep over expansion factors (8x, 32x) and sparsity levels $k \in 25, 50, 75, 100, 200$ across all layers. Input activations are normalized to unit norm for stable training and metric comparability across layers.

Figures 8–10 summarize the results. We observe a clear trade-off between sparsity and reconstruction quality (Fig. 1), with little difference between 8x and 32x expansions, with 8x performing even better for HuBERT under high sparsity (low L_0). While neuron survival decreases with higher expansion, the total number of active neurons still grows, staying at least twice the base model's size (Fig. 8). Notably, this ratio shows only weak correlation with L_2 quality, indicating that smaller values correspond to a suboptimal size–quality balance. Finally, Figs. 10 and 1 demonstrate that $k = 50$ with 8x expansion provides the best compromise between reconstruction fidelity, sparsity, and compression efficiency, minimizing both memory cost and inactive features.

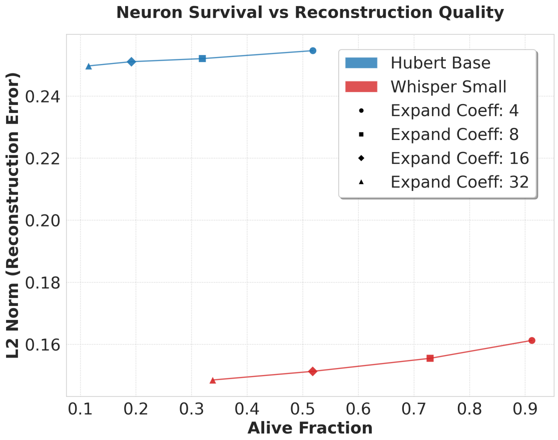


Figure 9: Connection between reconstruction quality and neurons survival rate

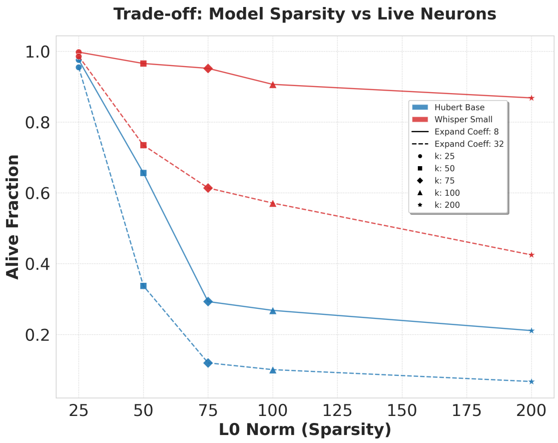


Figure 10: Connection between sparsity and neurons survival rate

B Feature robustness

For this set of experiments, we use datasets LibriSpeech, FSD and MTG for analysis of feature similarity (formulas 3.3.1 and 3.3.1). From these datasets we sample 500 audios with total of $749450, 74700$ and 86220 frames, respectively. We use BatchTopK with $k = 50$ on inference, meaning that for a single input audio of n frames, top $50 \times n$ of all $6144 \times n$ features will be denoted as active. For feature coverage we take Intersection-over-Union threshold $\theta = 0.5$, meaning that features with $\chi(a, b) > 0.5$ (see formula 3.3.1) are considered similar.

See results in tables 4 and 5.

C Domain-level feature specialization

We characterize each feature by two activation metrics: *activation frequency* and *average non-zero activation value*, computed at both frame (per token) and audio (per sample) levels for datasets representing three predefined domains: *speech*, *sounds*, and *music*.

Model	DS	L1	L4	L7	L10	L12
	LS	419	1768	2930	4295	3164
Hub_Hub2	FSD	447	1395	2292	2725	1826
	MTG	363	1517	2537	3390	2267
	LS	1532	2495	3637	5049	4340
Hub_Hub100K	FSD	1544	2220	2956	3314	2919
	MTG	1672	2467	3464	4396	4009
	LS	92	258	453	1466	
Hub_Hub$_{L+n}$	FSD	131	171	146	263	
	MTG	97	184	122	242	
	LS	1277	1987	1610	3746	4650
Wh_Wh100K	FSD	1354	1931	1550	2816	2606
	MTG	1888	2598	1923	3653	3783
	LS	921	1296	634	2692	
Wh_Wh$_{L+n}$	FSD	960	1267	629	1032	
	MTG	1420	1748	583	1849	
	LS	50	95	65	125	180
Hub_Wh	FSD	25	25	12	11	13
	MTG	44	29	15	10	6

Table 4: SAE feature set coverage between models and layers. LS (LibriSpeech), FSD (FSD50K) and MTG mean datasets for coverage score calculation. Suffix ^{2} is for SAE trained on the same activations but initialized with different random seeds; suffix 100K is for early stage of SAE training (100K iterations); suffix $_{L+n}$ is for coverage between different layers of the model (each layer is compared with the layer from the next column, i.e. in L4 column we show the coverage between features from layer 4 with features from layer 7).

Model	DS	L1	L4	L7	L10	L12
	LS	27	47	102	352	219
Hub	FSD	122	82	66	101	171
	MTG	33	30	22	42	56
	LS	23	49	88	367	232
Hub100K	FSD	66	71	47	113	132
	MTG	19	25	41	45	60
	LS	40	45	88	355	227
Hub2	FSD	112	72	57	94	135
	MTG	38	34	23	53	68
	LS	793	755	787	221	230
Wh	FSD	878	947	733	306	286
	MTG	1368	1328	736	101	65
	LS	844	718	736	144	166
Wh100K	FSD	962	868	608	191	135
	MTG	1470	1375	754	91	69

Table 5: Numbers of features having duplicates within same SAE. Dataset and model names are as in Table 4

For each domain combination (e.g., [speech, sounds, music], [speech, sounds], [music, sounds]), features are assigned to domains through threshold-based comparison. Specifically, for each feature j and domain combination c, we compute the activation frequency $f_{i,j}$ for each domain i in that combination. A feature is assigned to domain i^* if:

$$i^* = \arg\max_i f_{i,j} \quad \text{and} \quad \forall k \neq i^* : f_{i^*,j} - f_{k,j} \geq \tau$$

where τ is a threshold from the progressive threshold set. Assignment occurs when the maximum activation frequency exceeds all others by at least τ, providing graded confidence levels based on the threshold used. Features that fail to meet any threshold are marked as *unassigned*, while inactive ones (with $f_{i,j} = 0$ for all domains) are labeled *dead*. This procedure yields categorical labels and frequency-weighted color codes for visualization, with color intensity modulated by the threshold index k:

$$\text{color}_j = \text{RGB}_{\text{base}} \times (1 - c_{\text{coeff}} \cdot k), \quad c_{\text{coeff}} = 0.2$$

Final labels are aggregated across all domain combinations (three-way and pairwise), ensuring consistent categorization across contexts. The resulting assignments are visualized using t-SNE projections of SAE encoder weights, with colors corresponding to final domain labels. We additionally construct Venn diagrams to quantify overlap and exclusivity and to track the distribution of specialized features across model layers.

C.1 Experimental setup

Datasets. Seven datasets were used, grouped into three primary categories:

- **Speech:** LS-test-clean, IEMOCAP , ESD, Expresso, MELD, Demand
- **Sounds:** WHAM!, FSD50k, VocalSound, Nonspeech7k, ESC-50, VGGSound
- **Music:** MTG-Jamendo

Thresholds. Progressive thresholds were applied to ensure robust, confidence-graded specialization:

- Frame-level: $\tau \in \{0.2, 0.1, 0.04\}$
- Audio-level: $\tau \in \{0.5, 0.3\}$

Frame-level thresholds identify fine-grained feature specialization across individual tokens, while audio-level thresholds target coarser patterns observable at the sample level.

Audio-level analysis. Audio-level domain specialization captures features responsive to global acoustic properties and long-range dependencies. Thresholds of 0.5 and 0.3 were selected to emphasize features with substantial full-sample activation while maintaining discrimination between domains. This level of analysis complements frame-level detection by revealing features whose specialization is consistent across entire audio samples rather than transient in individual tokens.

Detection across multiple domain combinations (**[speech, sounds, music]**, **[speech, sounds]**, **[speech, music]**, **[sounds, music]**) allows disambiguation of overlapping feature roles that remain hidden in single-domain analysis. Formally, this multi-combination strategy leverages pairwise comparisons to identify features salient for two domains but not the third, enabling detection of subtle cross-modal relationships. For instance, features activated by both speech and environmental sounds but not music are properly identified in the [speech, sounds] experiment while remaining unassigned in the comprehensive three-domains analysis.

C.2 Frequency analysis

We present frequency-based domain specialization analysis across Whisper and HuBERT layers 6 and 7 (Fig. 11), two depths where acoustic feature learning remains interpretable yet incorporates substantial linguistic context. Scatter plots of activation frequency (x-axis) versus average non-zero activation value (y-axis) reveal distinct frequency distributions, domain-specific patterns, and model-dependent differences in activation magnitude.

Audio-level frequency exhibits a markedly different distribution than frame-level frequency. Features in both Whisper and HuBERT span the full frequency range from $f_{i,j}^{\text{audio}} = 0$ (never activated in any samples from any domains) to $f_{i,j}^{\text{audio}} = 1.0$ (activated in all samples from every domain). This coverage reflects the aggregative nature of audio-level frequency: features that activate sparsely at the frame level may still reach high sample-level frequency if their activations are distributed across many different samples.

Whisper exhibits pronounced clustering of music features (Fig. 11, top) at high average activation values ($\approx$ 3–6) with frequencies spanning 0.1 to

0.2 and 0.25 to 0.4. Speech (red) and sounds (blue) features are at lower activation values (≈ 0.5–2) with frequencies distributed across the full range. This pattern suggests that Whisper has learned a dedicated set of music-responsive features with high activation magnitude.

HuBERT exhibits a different audio-level profile (Fig. 11, bottom): the feature space shows dense, undifferentiated specialization at low average activation values (≈ 0.4–1.0) across all frequencies up to 1.0. Domain specialization is visible through domain colors (red (speech), blue (sounds), green (music)) but without any separation in activation magnitude observed in Whisper. This suggests HuBERT distributes specialization across more features with comparable activation strengths.

Frame-level frequency exhibits differences from audio-level analysis, with maximum frequencies typically not exceeding $f_{i,j}^{\text{frame}} \approx 0.5$, and substantial feature specialization at frequencies near zero. This sparsity pattern reflects the discrete, context-dependent nature of feature activation in sparse models: a feature may activate in only a small fraction of frames within a domain, even if it appears in most samples at the audio level. Whisper's and HuBERT's frame-level distribution are compressed, with most features concentrated in $f^{\text{frame}} < 0.1$.

C.3 Encoder matrix decomposition analysis

To directly probe how domain-specialized features are organized in representation space, encoder matrix decomposition is applied to the SAE encoder weights corresponding to layers 6 and 7 of Whisper and HuBERT. This analysis operates on a filtered subset of features, selected using the same activation-frequency statistics as in Section C.2 to ensure that only active and interpretable units are retained. Only those features are used that have been applied at least once to any domain in any domain combination.

The filtered encoder weight matrix $W_{\text{enc}} \in \mathbb{R}^{M \times d}$ (with $M = 8 \times 768$ latent units for Whisper and HuBERT) is then projected to two dimensions using t-SNE.

Each point in the resulting 2D embedding corresponds to a single SAE feature and is colored according to its domain assignment from the threshold-based procedure. Then for every domain combination, for example, for the combination [speech, sound, music] the features active for this combination are colored, while unassigned gray features will be active for one of the combinations, e.g. for [speech, sound]. Also, brighter dots reflect features with greater frequency differences.

The t-SNE decomposition of the SAE encoder matrix is presented for audio-level setup for Whisper layer 6 (see Fig. 22) and HuBERT layer 6 (see Fig. 23).

C.4 Multi domain features analysis

Feature specialization overlap is quantified through set-theoretic analysis. Define $\mathcal{S}_D = \{j : \text{feature } j \text{ assigned to domain } D\}$ for each domain $D \in \{\text{speech}, \text{sounds}, \text{music}\}$. Venn diagrams visualize sets $|\mathcal{S}_D|$ and all pairwise intersections, revealing the structure of cross-modal feature dependencies. The main observation is that for most layers for both models the speech set is separated from the sound and music sets, and the sound set is almost completely absorbed by the music set at both the audio and frame levels. See Fig. 12.

Those features that are strictly related to speech, sounds, or music (lie outside the intersections) form a Fig. 2 for layer-wise analysis.

C.5 Layer-wise analysis

In addition to the main text (Section 3.3.2).

Lower proportion of Whisper's speech features (compared to music) may reflect either stronger compression of speech information or a more efficient internal representation for speech relative to music.

HuBERT shows more speech features at the audio level and lower at the frame level than Whisper. This pattern is consistent with the interpretation that HuBERT is more sensitive to global audio attributes, whereas Whisper contains a richer set of frame level features, related to local semantic information, what is caused by the difference in pre-training objectives and data composition.

Sound features (blue) are consistently underrepresented in both models: they are nearly absent in HuBERT and appear only sparsely in Whisper, primarily at the frame level. An explanation is that many sound features co-occur more frequently with music than with the sound domain, causing the sound feature set to be effectively subsumed by music-associated activations. This conclusion is supported by an additional Fig. 12.

D Classification

The following datasets are selected for the classifiers training: 5000 audios from LibriTTS (Zen

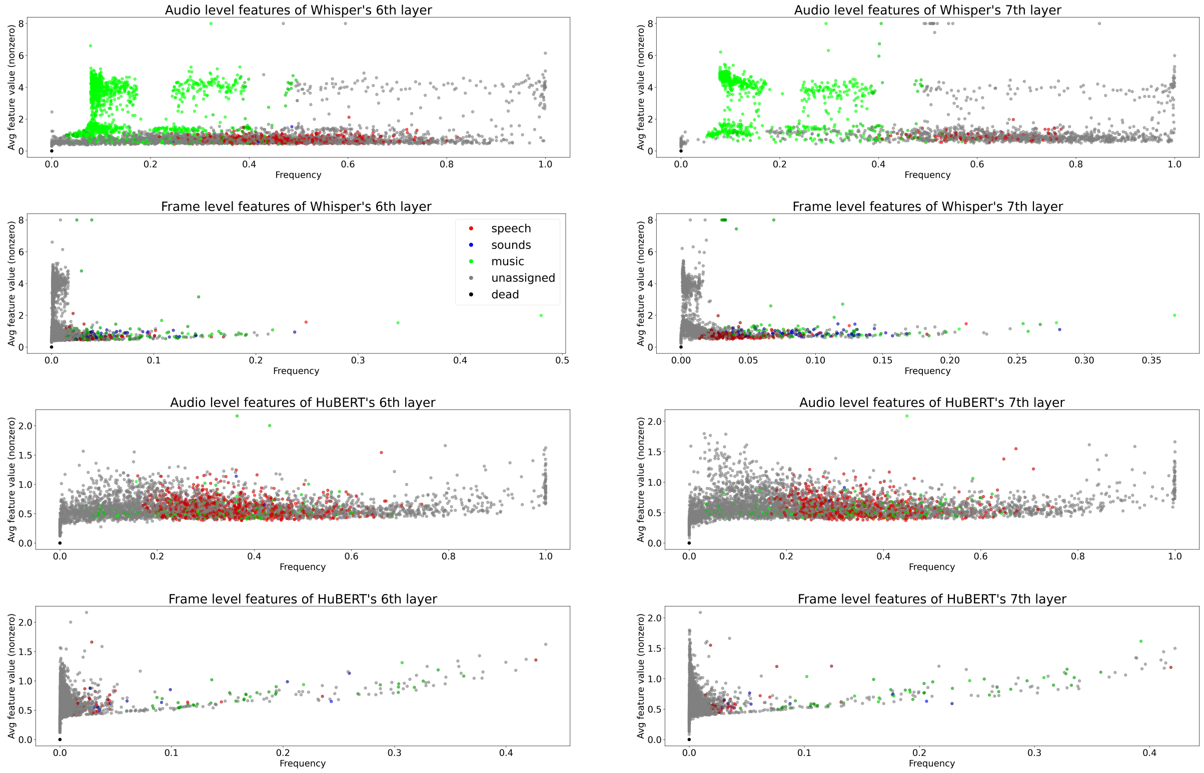


Figure 11: **Frequency-based domain specialization.** Audio-level (1st and 3rd rows from the top) and frame-level (2nd and 4th rows from the top) activation frequency versus activation magnitude for Whisper (1st and 2nd rows from the top) and HuBERT (3rd and 4th rows from the top) layers 6–7 (left and right columns respectively). Colors: red (speech), blue (sounds), green (music), gray (unassigned), black (dead).

et al., 2019) dataset for gender classification; 2500 clean and 2500 speech samples from Demand dataset; 1500 for each of five accents – American, British, Indian, Irish, Scottish – from VCTK for accent classification; and English part of ESD dataset for a 5 class emotion classification, encompassing the emotions angry, happy, neutral, sad, and surprise.

We used `LogisticRegression` class from `scikit-learn` with parameters `max_iter=10000, penalty='none', solver='newton-cg'`.

All results are presented in Fig. 13

E Vowel unlearning details

E.1 Technical details

In our unlearning experiments, we iteratively removed SAE features in order of their discriminative power (estimated by Fisher score) for a particular spoken letter (vowel) and retrained a LogisticRegression classifier after each removal to measure vowel recognition performance on the remaining features. This allows us to track whether the SAE embeddings still retain information about each vowel class. We employed two distinct regularization approaches to examine their impact:

Standard Regularization Setting: Following established practices (Jourdan et al., 2024; Ravfogel et al., 2022), we initially used `LogisticRegression` from `scikit-learn` with default hyperparameters. However, during preliminary experiments, we encountered convergence issues with the default 100 iterations, which we resolved by increasing `max_iter` to 10000. This `max_iter` value was maintained throughout all further experiments.

No Regularization Setting: We conducted our main experiments, featured in the Fig. 4, using `LogisticRegression` with no regularization (`penalty='none', solver='newton-cg'`) and `max_iter=10000`. We find this unregularized approach preferable, as it provides the most rigorous test of information erasure by allowing the classifier to fully exploit any remaining information in the features without the artificial constraints imposed by regularization.

Both experimental settings employed a 5:2 train/test split with stratification across speakers and letters, ensuring balanced representation

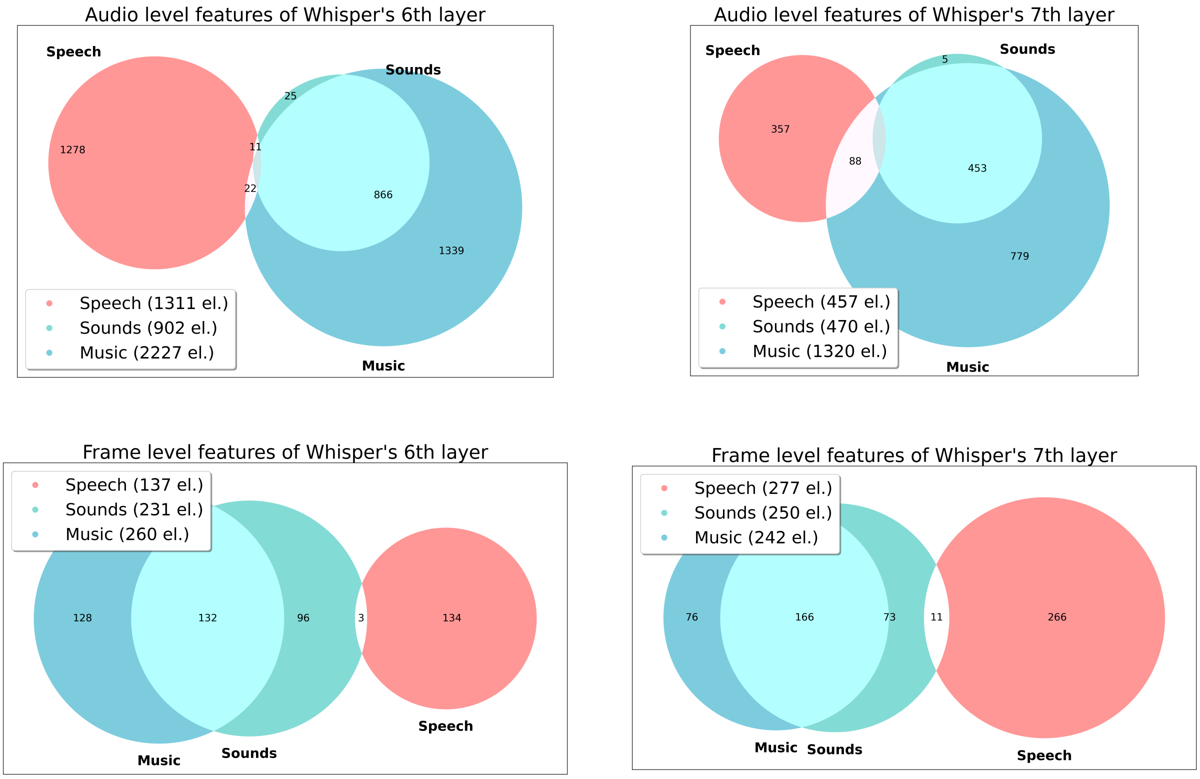


Figure 12: **Feature overlap** for Whisper (layers 6 and 7): Venn diagrams for audio and frame levels.

and preventing bias toward specific speakers or phonemes.

E.2 Unlearning plots for various letters and regularization setups

Fig. 24 and 25 present vowel unlearning experiments for HuBERT's 12th layer using standard `LogisticRegression` with default `l2` regularization, default value `C=1`, and increased `max_iter=10000`. For experiments without regularization (`penalty='none'`) see Fig. 26 and 27.

These experiments reveal a significant difference: unregularized logistic regression requires removal of over 1000 features for successful unlearning, while logistic regression with standard L2 regularization achieve comparable results with only 160–400 features (3–6% of the total). This is even fewer than the features required when working with original HuBERT activations.

However, we caution that these L2 regularization results may be overly optimistic: regularization artificially constrains the classifier's capacity to extract information, potentially masking the presence of recoverable information rather than confirming its absence. Unregularized classifiers, which can fully exploit all available patterns, provide a more realistic assessment of true information removal, suggesting that genuine unlearning requires the more extensive feature removal observed in our unregularized experiments. In the same time, unregularized classifiers have their own limitations in our experimental setting: since we have more features than training samples, they exhibit poor convergence and numerical instability, making them challenging to work with despite providing more rigorous tests of information erasure.

E.3 k-probing vowels

In addition to the experiments where we progressively removed features, we ran a "reverse" series of tests in which features were added one by one—starting with the most informative according to the Fisher score and then adding less important ones. We discovered that to regain high accuracy in recognizing a single vowel against the others, it was enough to activate just one or two of the top-ranked features in both SAE activations and HuBERT embeddings (see Fig. 28. This indicates that the highest Fisher-ranked features carry enough of the phonetic information for a reliable

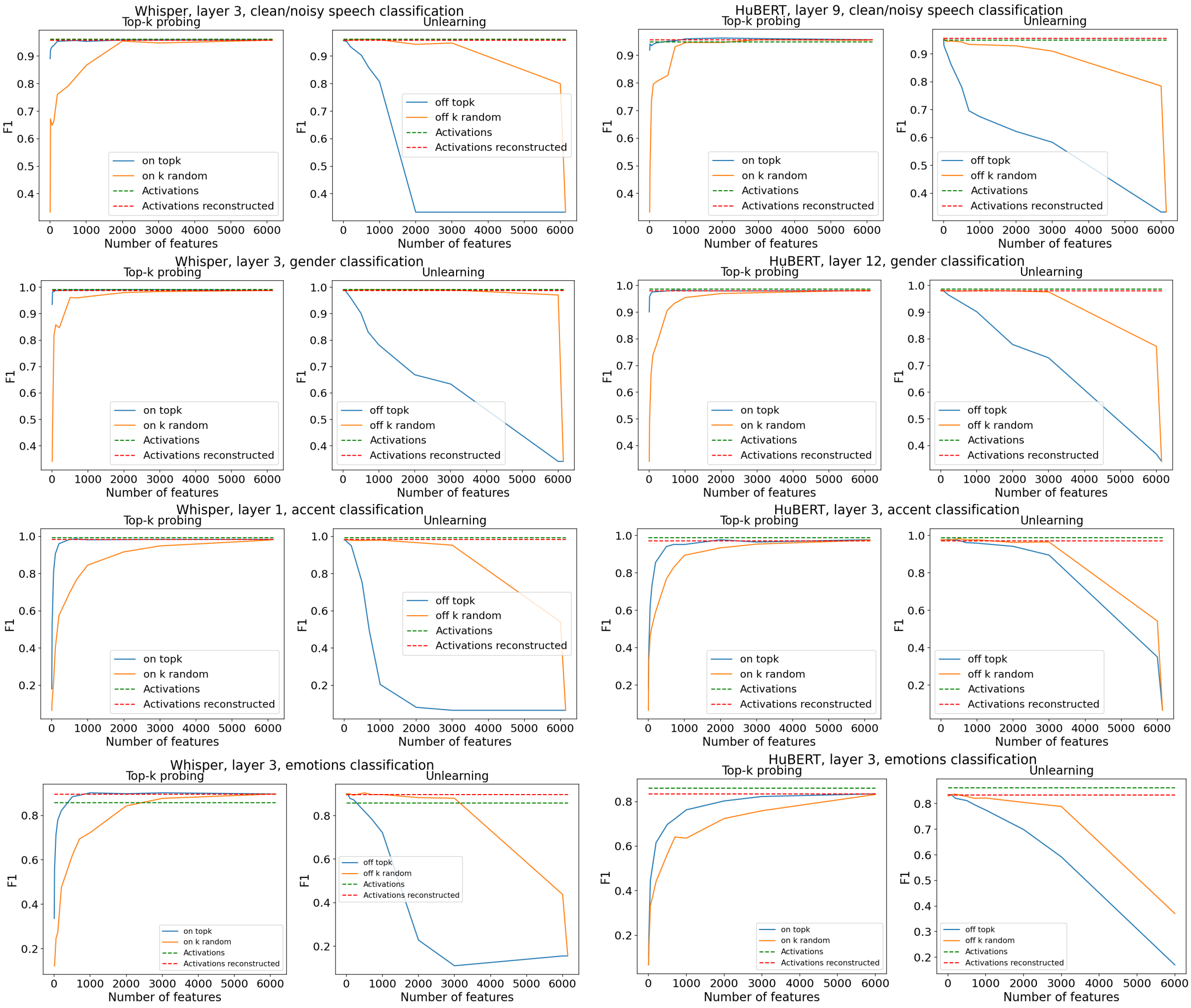


Figure 13: Top-k probing and unlearning for four classification tasks

classification. For this experiment we also used Logistic Regression without regularization.

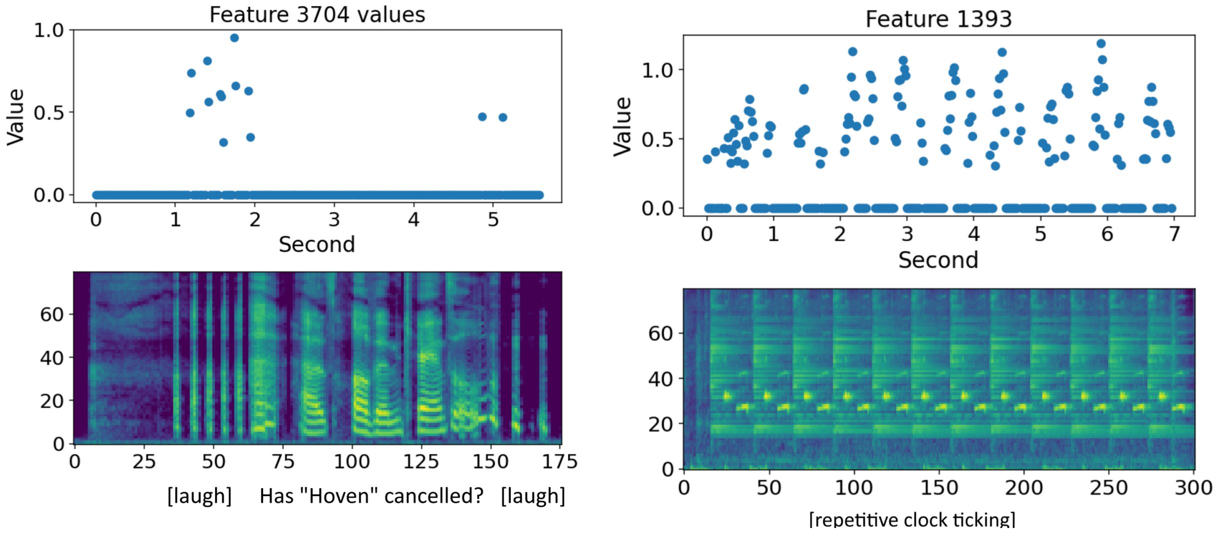
F Interpretation by labels

The feature search procedure is as follows: (1) identify all latents that are activated on samples with the target labels; (2) for each latent, evaluate the F1 score at different thresholds in steps of 0.1 across the interval from its minimum to maximum activation value; (3) if the F1 score at any threshold exceeds 0.5, consider that feature correlates with the target label.

Fig. 14 presents some interesting features found during the experiment and may be illustrated by mel-spectrogram representation and the feature activated frames. The corresponding audio fragments are available on the demo page.

G Auto-interpretation details

Fig.16 presents a word map of characteristic labels, where for both Hubert and Whisper models, the dominant interpretation is related to speech. However, the following limitations should be considered. First, the threshold value of 0.1 was empirically selected. Second, the test data set consists largely of speech data. This bias toward speech may cause small but frequent activations of features, pushing the resulting label toward a speech interpretation and obscuring rarer music or sound events. Additional limitation of this method is its dependence on the capabilities of the underlying audio captioning. This is particularly evident when interpreting phoneme-level features. For instance, a feature responsible for the vowel sound "A" will produce audio chunks consisting of various people producing that isolated sound. This lack of broader acoustic context often confuses the captioning model, which may then misclassify the sound



(a) HuBERT, layer 4, "laughter"

(b) HuBERT, layer 4, "repetitive sounds (e.g. siren, ticking)"

Figure 14: Additional features found in classification by label experiment.

as generic multi-speaker dialogue rather than identifying the specific phoneme.

H Steering details

As a baseline we propose to mitigate hallucinations in the Whisper speech recognition model by applying steering vectors to its internal activations. The steering vector is derived by contrasting latent representations of hallucinatory and non-hallucinatory samples, with labels automatically assigned based on Whisper's internal no-speech probability score.

The core hypothesis is that a direction in the activation space can be identified that, when amplified, suppresses the model's tendency to generate spurious transcriptions for non-speech audio inputs.

We compute the normalized difference between mean activations:

$$\vec{s} = \frac{\overline{\text{act}_H} - \overline{\text{act}_N}}{\|\overline{\text{act}_H} - \overline{\text{act}_N}\|},$$

where H and N are Hallucinations and Non-hallucinations clusters respectively, where H is represented by non-speech samples with no_speech_prob $< \tau$ and N by non-speech samples with no_speech_prob $\geq \tau$.

H.1 Experiment setup

Datasets: Experiments are conducted on three non-speech datasets: FSD50k (sound events), Musan (general noise), WHAM (noisy speech without intelligible speech). For FSD50k samples with speech related lables are filtered. To evaluate the impact on genuine speech recognition performance, we use the LibriSpeech test-clean dataset.

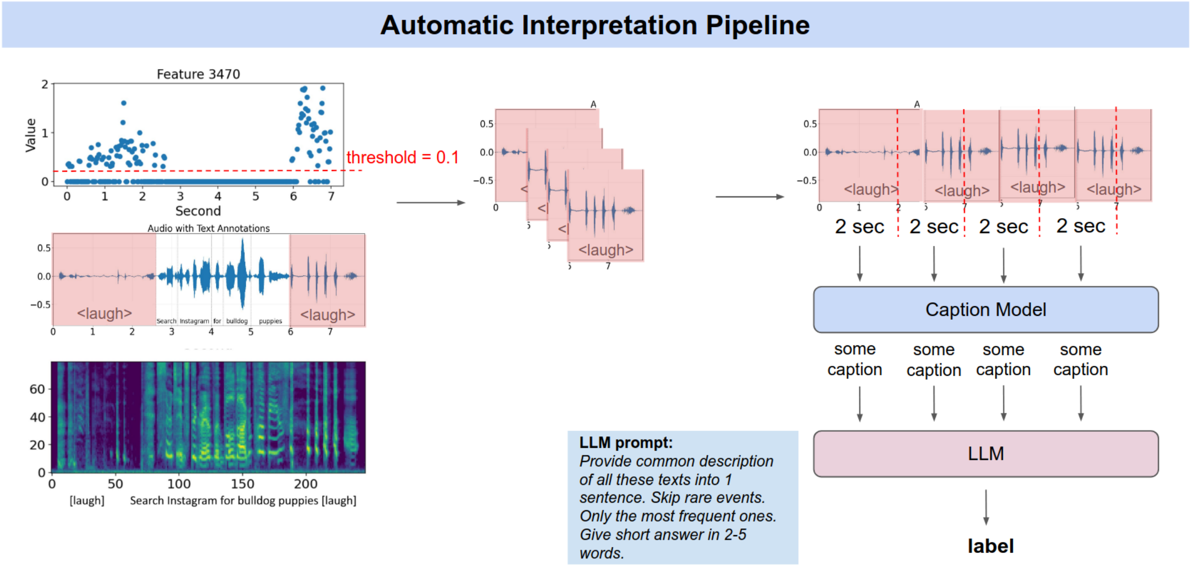
Model: All experiments are based on the Whisper small model on activations after transformer block of AudioEncoder after 8th layer.

Metrics: Our primary metric for evaluating hallucination reduction is the False Positive Rate (FPR), defined as the proportion of non-speech audio clips for which the model generates any transcription with a no_speech_prob below a set threshold equals to 0.5. We also report the standard Word Error Rate (WER) on LibriSpeech to ensure that steering does not degrade performance on legitimate speech tasks. Due to the fact that after steering, the no_speech_prob parameter distribution on LibriSpeech dataset practically did not shift to the right, WER is a much better estimate of the preservation of the model's ability to recognize speech than True Positive Rate (TPR) or AUC score.

H.2 Results and visualization

Identifying the SAE features responsible for hallucinations was accomplished through a classification task using logistic regression. Calculated F1 metric depends on the hyperparameter k (the number of SAE features used in the classification). This allows us to find a tradeoff between classification accuracy and the number of SAE features used. Our intuition was that although hallucinations are a complex concept, we want to find the minimum k with quality at the level of the entire SAE vector classification. The results are presented in Table 7.

For the baseline configuration, we pursued a dual



LLM prompt:
Provide common description of all these texts into 1 sentence. Skip rare events. Only the most frequent ones. Give short answer in 2-5 words.

Figure 15: Automatic Interpretation Pipeline

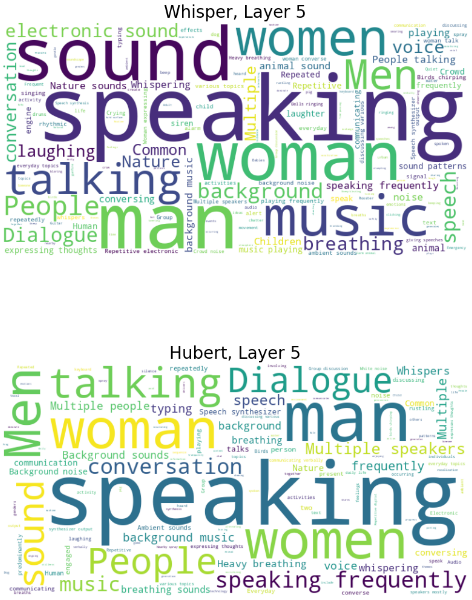


Figure 16: Features label frequencies for Whisper and Hubert models.

optimization objective: tuning the hyperparameter α while simultaneously identifying the dataset that yields the most effective steering vectors. Steering vectors were calculated independently for each dataset using its corresponding activation distributions. Each dataset's steering vector was then evaluated across a range of α values, applied to all datasets to assess both the hyperparameter sensitivity and the generalization performance of vectors originating from different source datasets. The results clearly show that the best steering vector is obtained on the Musan dataset with $\alpha = 3$, as presented in Table 8.

To verify that the proposed steering vectors do not degrade standard ASR performance, we evaluate them on the LibriSpeech test-clean set for several values of the steering strength α. For each steering configuration we measure the Word Error Rate (WER) of the ASR model. The results are summarized in Table 9. For Musan, FSD50k and WHAM steering vectors, WER remains essentially unchanged with respect to the unsteered model (around 0.05) across all tested values of α, indicating that these steering directions do not harm recognition quality on clean speech.

Whisper inference with and without SAE and the effect of SAE on FPR are examined separately. Table 12 shows that the addition of SAE does not significantly shift the distribution of the parameter no_speech_prob in the Musan and FSD50k datasets, but on the WHAM dataset, the FPR decreases from 0.51 to 0.36. This phenomenon requires further study. Furthermore, inference with SAE does not significantly change WER, as shown in Table 11.

Unlike baseline hyperparameter optimization experiments, SAE-based steering introduces an additional hyperparameter k, the number of SAE features whose activations are steered. These features are selected according to their importance in the hallucination classifier. Thus, SAE steering requires jointly choosing both the scaling factor α and the sparsity level k. Table 10 reports the resulting FPR for each steering vector on each evaluation dataset. Experiments shows that a steering vector constructed on the FSD50k dataset with $k = 100$ and $\alpha = 3$ drives the FPR close to zero on all evaluation datasets. However, steering should not only suppress hallucinations but also preserve recognition quality. Therefore, we highlight two configurations: an *extreme* setting, which achieves the strongest hallucination suppression with $k = 100$ and $\alpha = 3$, and an *optimal* setting, which uses the same number of features but a milder scaling, $k = 100$ and $\alpha = 1$, to better balance hallucination reduction and ASR accuracy.

An analysis of the impact of SAE steering on speech recognition quality is also presented in Table 13, which shows that *extreme* steering significantly degrades the model's ability to perform its original task. Meanwhile, *optimal* steering degrades WER by only 0.3%, while reducing FPR by 70% (0.37 -> 0.11).

The no_speech_prob parameter distribution shift after steering plots for the selected configurations (baseline Musan steering vector with $alpha = 3$ and SAE steering vectors with top-$k = 100$ and $\alpha \in \{1, 3\}$) are also presented. For the Musan (Fig. 18) dataset, FSD50k (Fig. 17) and WHAM (Fig. 19) FPR was calculated, for LibriSpeech test-clean (Fig. 20) TPR (higher is better) was calculated.

I Mel-interpretation details

The experiment was designed to find features which activates at the beginning or end of a word. It was assumed that the energy in the averaged feature maps from the mel spectrograms would be shifted to the right and left, respectively; that is, for a word-beginning feature, the energy would be to the right of center, and for a word-ending feature, it would be to the left. These maps were found among the top 5 features with the highest activation frequency

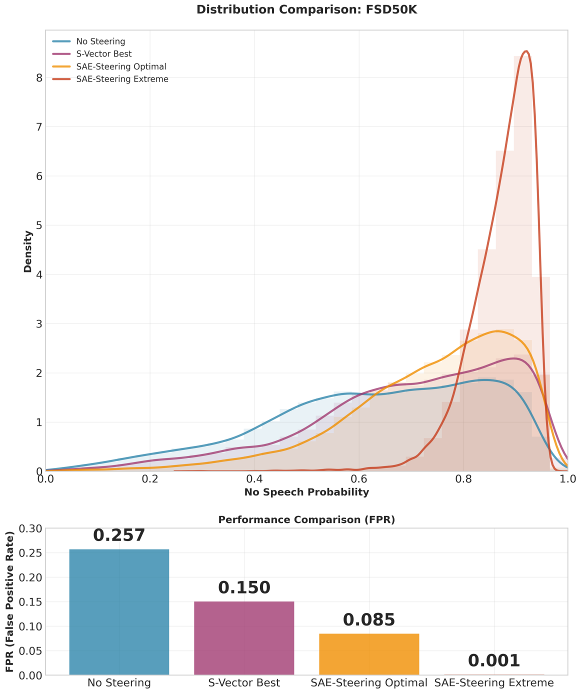


Figure 17: Distribution of no_speech_prob on the FSD50k dataset before and after applying steering vectors. The post-steering distribution is skewed towards 1.0.

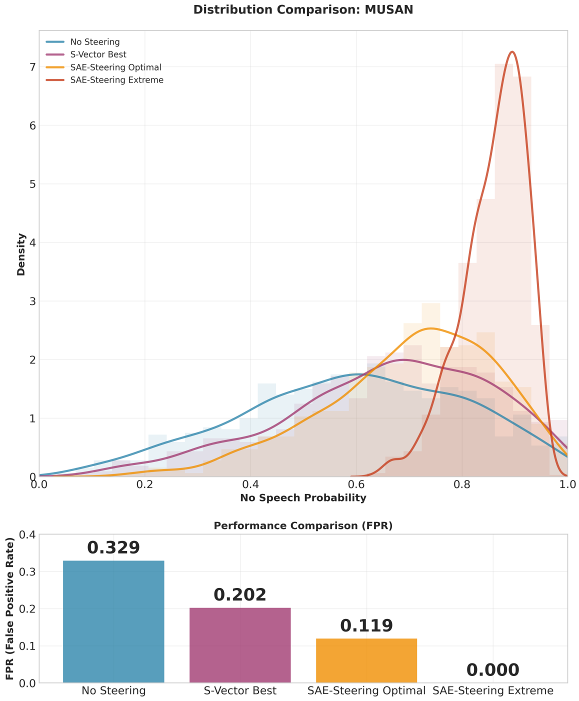


Figure 18: Distribution of no_speech_prob on the Musan dataset before and after applying steering vectors. The post-steering distribution is skewed towards 1.0.

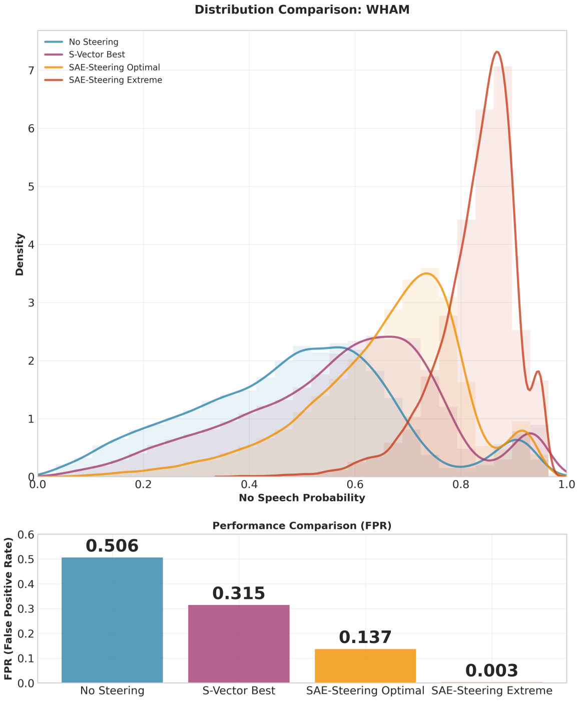


Figure 19: Distribution of no_speech_prob on the WHAM dataset before and after applying steering vectors.

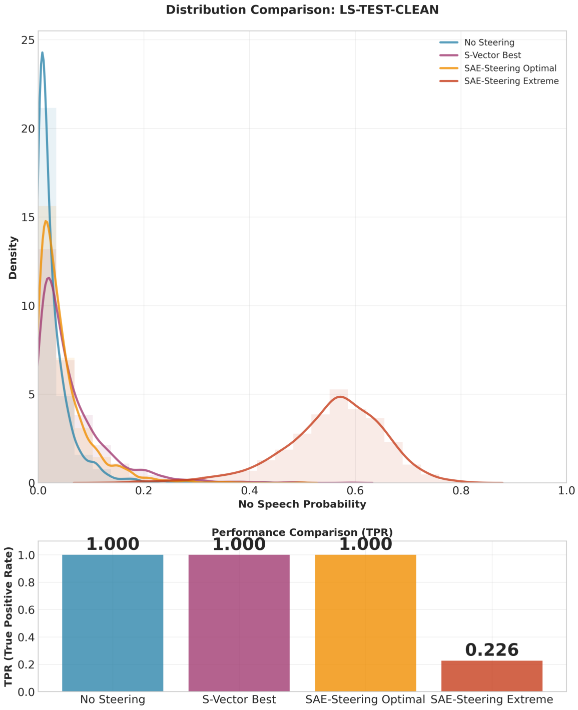


Figure 20: Distribution of no_speech_prob on the LibriSpeech test-clean dataset before and after applying steering vectors. The post-steering distribution is skewed towards 1.0.

difference between the speech domain and the music and sound domains, based on frequencies from Domain Specialization experiments. Therefore, it can be concluded that detected features are strongly speech-specific. Moreover, they were present in several layers of HuBERT and Whisper, both at the beginning and closer to the end of the network. However, only a variant for HuBERT layer 11 is presented in the article in Fig. 21.

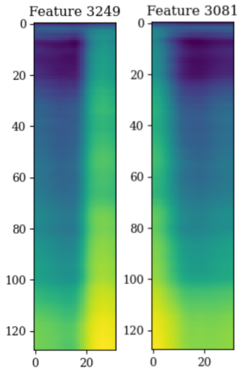


Figure 21: Features 3249 and 3081 of Hubert's SAE from layer 11.

J Details of EEG experiments

We chose data from midline parietal electrode Pz collected from 19 subjects listening to 5 excerpts from audiobook each 3 minutes long resulting in 15 minutes for each participant. SAE features (stimuli s) were extracted from these excerpts with models trained on the last 12-th layer of HuBERT-base and Whisper-base models before normalization. SAE features were normalized to have unit maximum whereas EEG signals (responses r) were first processed by band-pass filter keeping frequencies between 1Hz and 8Hz and then normalized to have zero median and unit interquartile range. Both EEG signals and SAE features were resampled to 128Hz. Temporal-response functions (TRFs) were built with mTRFpy Python package.

We randomly chose 1000 HuBERT and 1000 Whisper features activating at least once per second on average. For each feature f, we found time lag values $\tau_{min}^{(f)}$ and $\tau_{max}^{(f)}$ minimizing and maximizing TRFs respectively on the development set with the total duration of 6 minutes. Then, for each feature f for TRFs built on the test set with the overall duration of 9 minutes we performed two one-sided t-tests to check whether corresponding TRFs have statistically significant negative correlation at $\tau_{min}^{(f)}$ and positive correlation at $\tau_{max}^{(f)}$. After that, we applied Holm-Bonferroni correction to process the results of multiple statistical tests at significance level 0.05. As a result of this procedure, we found around 1% of Whisper and 1.5% of HuBERT features having significant correlation with Pz electrode response at certain time lags.

As one can see from Fig. 7, correlation between SAE and EEG features can occur with almost zero time lag. It may seem counterintuitive since it must take some time for brain to process audio information, but it has to be noted that HuBERT and Whisper feature extractors have access to both left and right audio context and can activate, for example, in the end of a particular word or sound, which can explain this seeming contradiction.

top-k	α	**Musan**	**FSD50k**	**WHAM**
Musan vec.				
1	0.5	0.31	0.29	0.30
1	1.0	0.31	0.29	0.30
1	3.0	0.31	0.24	0.30
5	0.5	0.29	0.29	0.31
5	1.0	0.27	0.29	0.31
5	3.0	0.20	0.24	0.29
10	0.5	0.29	0.29	0.31
10	1.0	0.27	0.27	0.30
10	3.0	0.20	0.19	0.25
100	0.5	0.27	0.21	0.30
100	1.0	0.20	0.12	0.26
100	3.0	**0.01**	**0.00**	**0.00**
FSD50k vec.				
1	0.5	0.24	0.23	0.24
1	1.0	0.24	0.23	0.24
1	3.0	0.24	0.20	0.23
5	0.5	0.23	0.23	0.24
5	1.0	0.21	0.23	0.24
5	3.0	0.16	0.20	0.23
10	0.5	0.23	0.23	0.24
10	1.0	0.21	0.21	0.24
10	3.0	0.16	0.14	0.19
100	0.5	0.20	0.15	0.23
100	1.0	0.16	0.09	0.20
100	3.0	**0.01**	**0.00**	0.01
WHAM vec.				
1	0.5	0.36	0.36	0.36
1	1.0	0.36	0.35	0.36
1	3.0	0.36	0.32	0.35
5	0.5	0.36	0.36	0.36
5	1.0	0.35	0.35	0.36
5	3.0	0.32	0.32	0.33
10	0.5	0.36	0.34	0.35
10	1.0	0.35	0.32	0.35
10	3.0	0.32	0.24	0.30
100	0.5	0.34	0.25	0.34
100	1.0	0.32	0.13	0.29
100	3.0	0.04	**0.00**	**0.00**

Table 6: False Positive Rate (FPR) for SAE steering with different configurations, full version of the Table 10. Tuning the scaling factor α and the number of top-k most informative SAE features selected from the hallucination classifier. Dataset specific columns represents the dataset on which FPR was calculated. Table is divided into 3 sections, each section refers to a specific dataset on which the SAE steering vector was formed.

	all	1000	100	10	5	1
Musan	0.72	0.74	0.67	0.55	0.42	0.33
FSD50k	0.72	0.74	0.73	0.55	0.51	0.35
WHAM	0.82	0.83	0.82	0.80	0.73	0.70

Table 7: F1 scores of the logistic–regression hallucination classifier as a function of the number k of SAE features used. Rows correspond to evaluation datasets (Musan, FSD50k, WHAM). Column *all* uses the full SAE vector, while *top-k* columns use only the k most informative SAE features, illustrating the tradeoff between classification quality and feature sparsity.

$\alpha = 0.5$

	No Steering	Musan	FSD50k	WHAM
Musan	0.33	**0.31**	0.32	0.34
FSD50k	0.27	**0.24**	0.25	0.27
WHAM	0.51	**0.47**	0.49	0.52

$\alpha = 1$

	No Steering	Musan	FSD50k	WHAM
Musan	0.33	**0.28**	0.31	0.34
FSD50k	0.27	**0.22**	0.24	0.28
WHAM	0.51	**0.44**	0.47	0.54

$\alpha = 3$

	No Steering	Musan	FSD50k	WHAM
Musan	0.33	**0.20**	0.28	0.36
FSD50k	0.27	**0.15**	0.22	0.30
WHAM	0.51	**0.32**	0.41	0.57

Table 8: False Positive Rate for baseline steering-vector steering at different α values. Columns show error when steering with the corresponding steering vector.

	No Steering	**Musan**	**FSD50k**	**WHAM**
$\alpha = 0.5$	0.051	0.051	0.051	0.051
$\alpha = 1.0$	0.051	0.051	0.051	0.051
$\alpha = 3.0$	0.051	0.053	0.055	0.053

Table 9: Steering vector validation on ASR task. Word error rate (WER; lower is better) of the Whisper-small model on LibriSpeech test-clean when steered with different steering vectors and strengths. Rows correspond to steering strength $\alpha \in \{0.5, 1.0, 3.0\}$, while columns indicate which steering vector is used.

Test	**Steering**	**Top-k=100**		**k=10**	**k=1**
Dataset	**Vector**	α=1.0	α=3.0	α=3.0	α=3.0
Musan	Musan	0.20	**0.01**	0.20	0.28
	FSD50K	0.12	**0.00**	0.19	0.31
	WHAM	0.26	**0.00**	0.25	0.34
FSD50K	Musan	0.16	**0.01**	0.16	0.22
	FSD50K	0.09	**0.00**	0.14	0.24
	WHAM	0.20	0.01	0.19	0.28
WHAM	Musan	0.32	0.04	0.32	0.44
	FSD50K	0.13	**0.00**	0.24	0.47
	WHAM	0.29	**0.00**	0.30	0.54

Table 10: False Positive Rate (FPR) for SAE steering with different configurations. Tuning the scaling factor α and the number of top-k most informative SAE features selected from the hallucination classifier. Test Dataset column represents the dataset on which FPR was calculated. Steering Vector column according to the dataset from which the SAE steering vector was formed. Short version, all experiments presented in the Table 6.

Method	**No Steering**	**SAE + No Steering**
Baseline	0.051	0.052

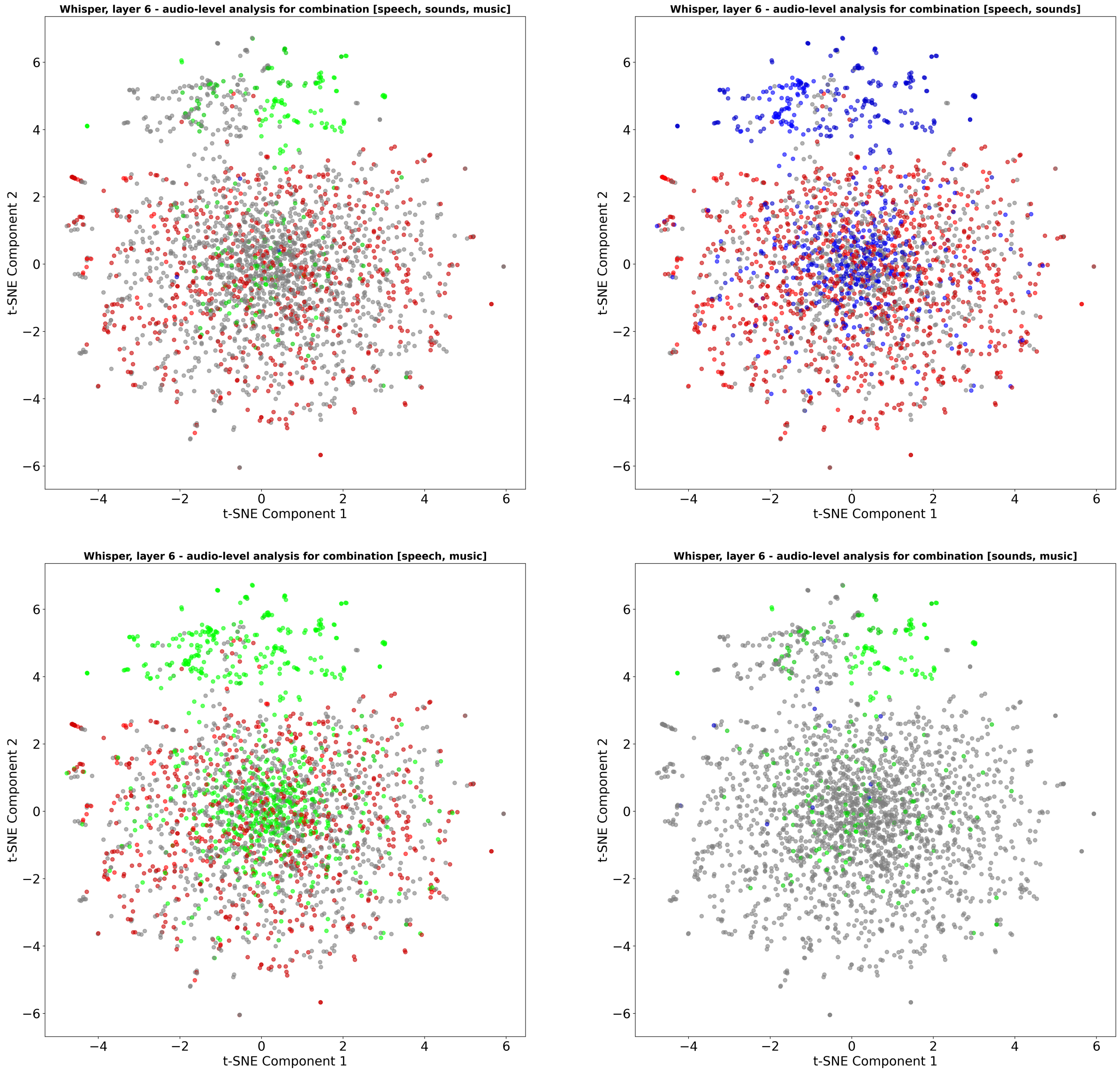
Table 11: Whisper small WER calculation on LibriSpeech test-clean without/with SAE.

Dataset	**No Steering**	**SAE + No Steering**
Musan	0.33	0.31
FSD50K	0.27	0.24
WHAM	0.51	0.36

Table 12: FPR Reduction in inference configurations without/with SAE.

		Word Error Rate (WER)			
S-Vector	α	$k = 100$	$k = 10$	$k = 5$	$k = 1$
	0.5	0.053	0.052	0.052	0.052
Musan	1.0	0.054	0.052	0.052	0.052
	3.0	0.139	0.054	0.054	0.052
	0.5	0.053	0.052	0.052	0.052
FSD50K	1.0	0.055	0.053	0.052	0.052
	3.0	0.984	0.054	0.052	0.052
	0.5	0.054	0.052	0.052	0.052
WHAM	1.0	0.064	0.052	0.052	0.052
	3.0	0.975	0.055	0.053	0.052

Table 13: WER on LibriSpeech test-clean when steering with SAE-based vectors. The table reports WER while jointly tuning the steering strength parameter α and the number of top-k most informative SAE features. Column S-Vector indicate the dataset used to calculate the SAE steering vector.



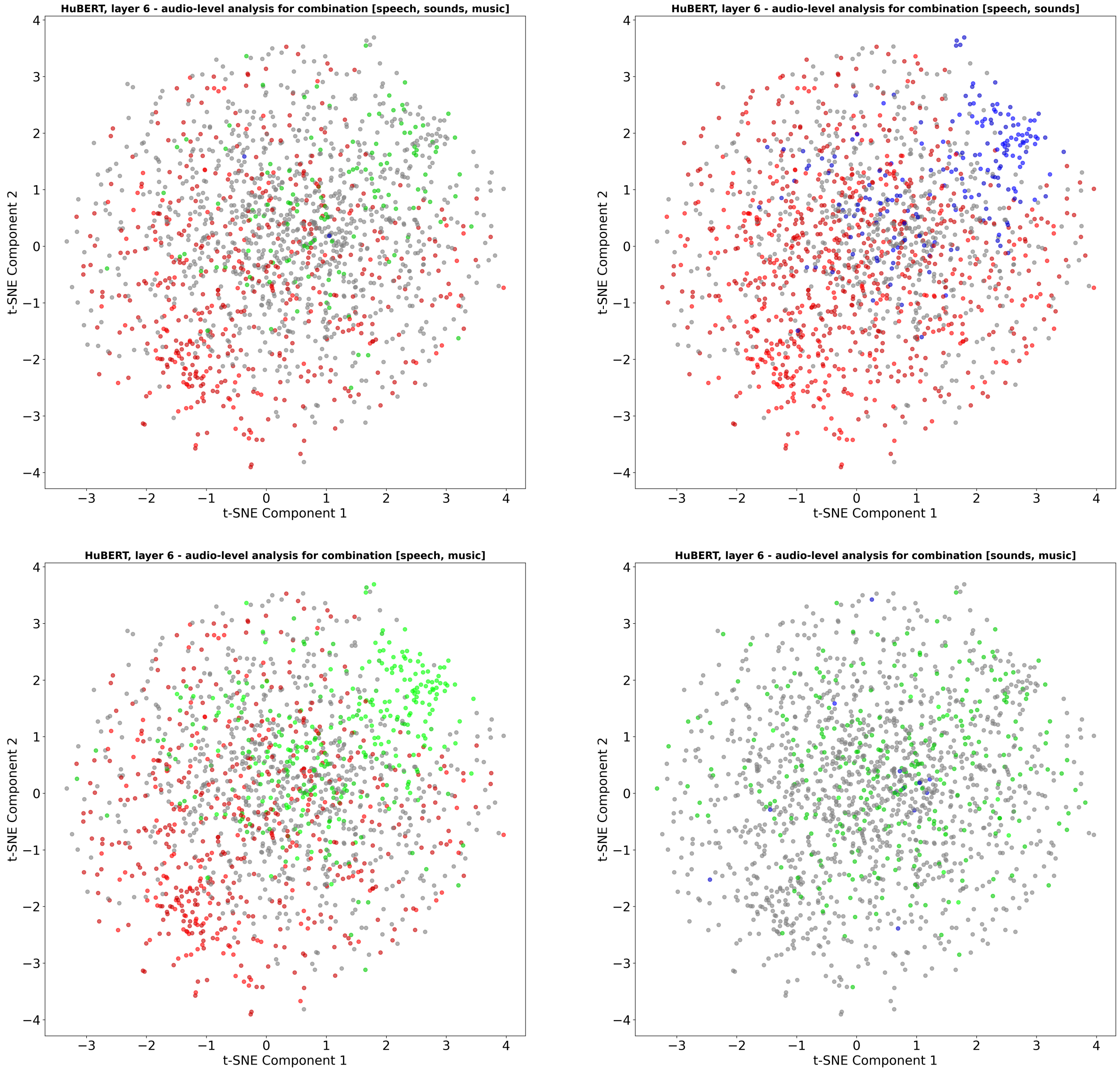
Whisper, layer 6 - audio-level analysis for combination [speech, sounds, music]

Whisper, layer 6 - audio-level analysis for combination [speech, sounds]

Whisper, layer 6 - audio-level analysis for combination [speech, music]

Whisper, layer 6 - audio-level analysis for combination [sounds, music]

Figure 22: t-SNE decomposition of SAE encoder weights for Whisper layer 6 in the audio-level setup. Each point corresponds to a single latent feature, colored by its domain assignment (speech, sounds, music, or unassigned) obtained from activation-frequency–based specialization. Brighter dots indicate features with larger activation frequency differences between domains, highlighting the most strongly specialized units in the representation space.


HuBERT, layer 6 - audio-level analysis for combination [speech, sounds, music]

HuBERT, layer 6 - audio-level analysis for combination [speech, sounds]

HuBERT, layer 6 - audio-level analysis for combination [speech, music]

HuBERT, layer 6 - audio-level analysis for combination [sounds, music]

Figure 23: t-SNE decomposition of SAE encoder weights for HuBERT layer 6 in the audio-level setup. Points represent individual latent features colored by their domain assignments, with unassigned gray features active only in alternative domain combinations.

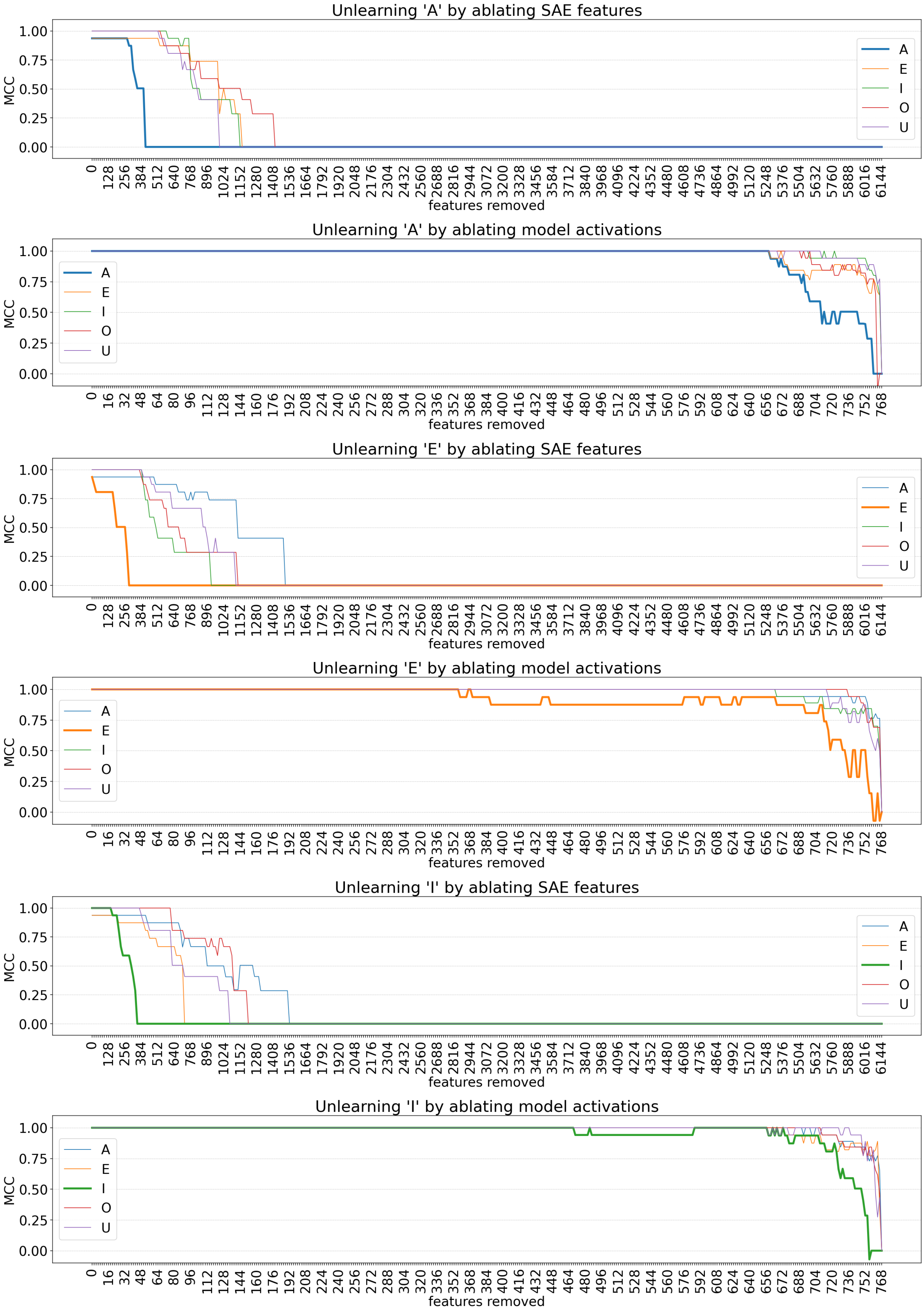


Figure 24: Unlearning plots for letters ’A’, ’E’ and ’I’ at the last layer of HuBERT model, using standard `LogisticRegression` with standard `penalty=’l2’` and `max_iter=10000`

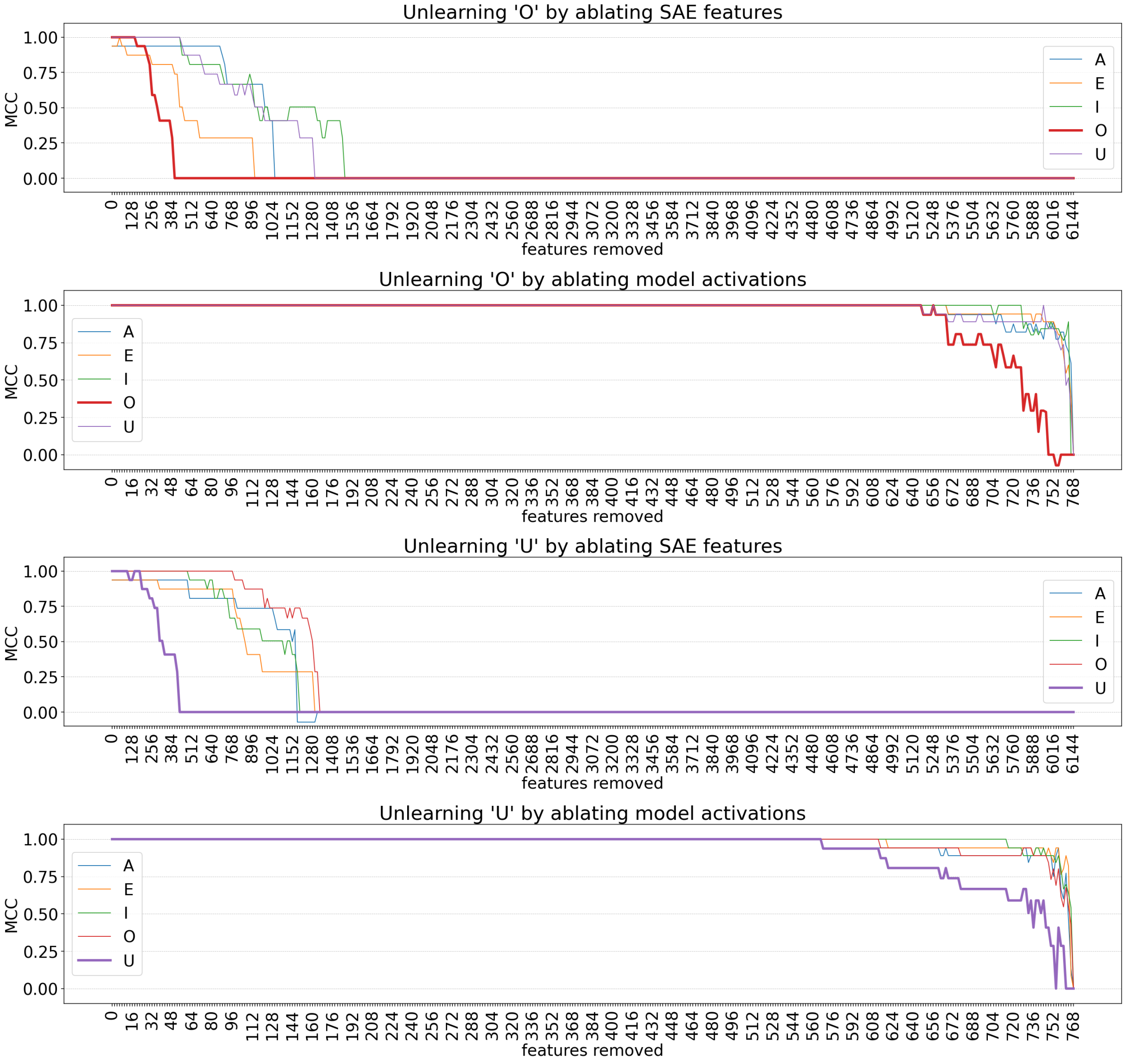


Figure 25: Unlearning plots for letters ’O’ and ’U’ at the last layer of HuBERT model, using standard `LogisticRegression` with standard `penalty=’l2’` and `max_iter=10000`

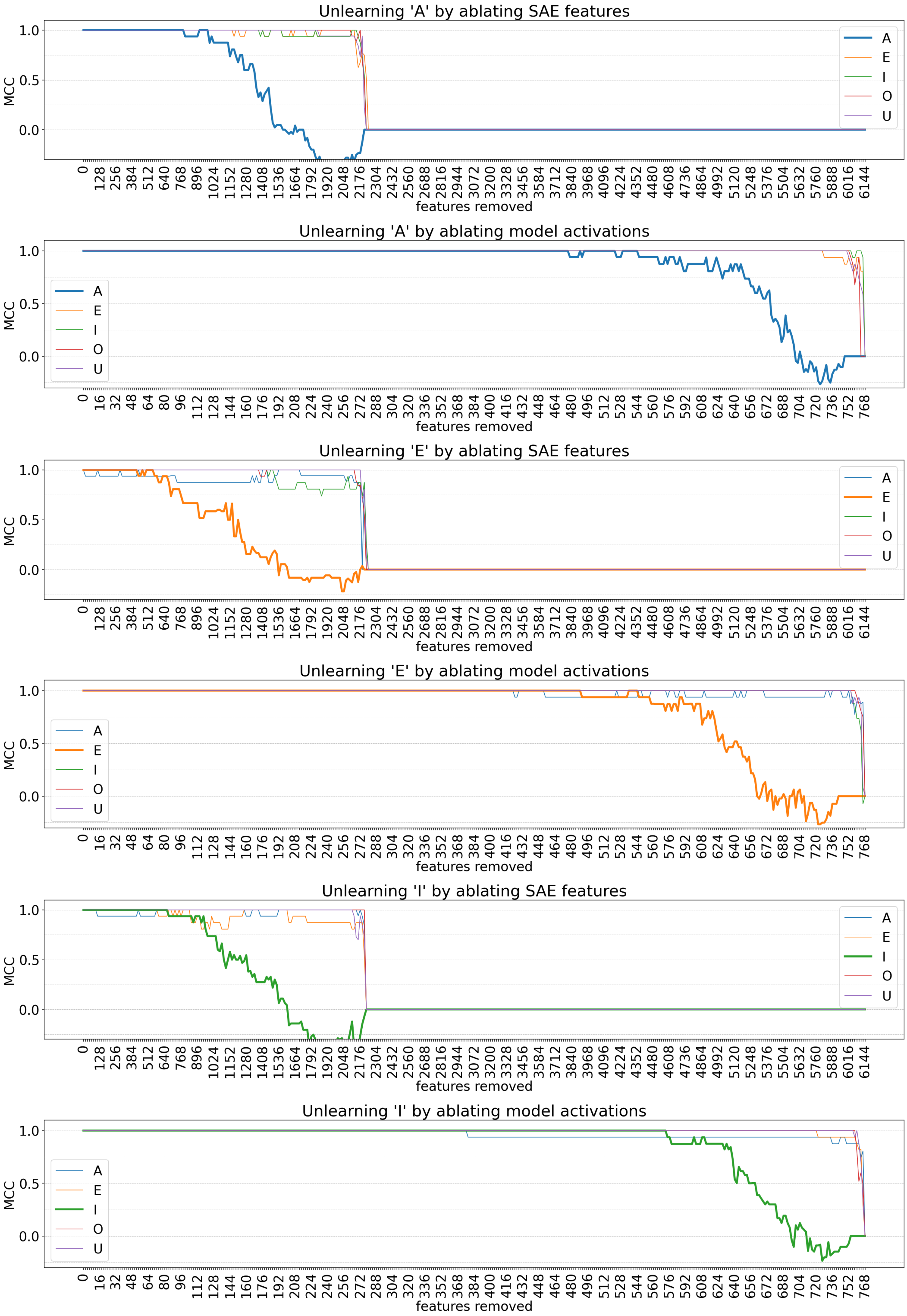


Figure 26: Unlearning plots for letters ’A’, ’E’ and ’I’ at the last layer of HuBERT model, using `LogisticRegression` without regularization and `max_iter=10000`

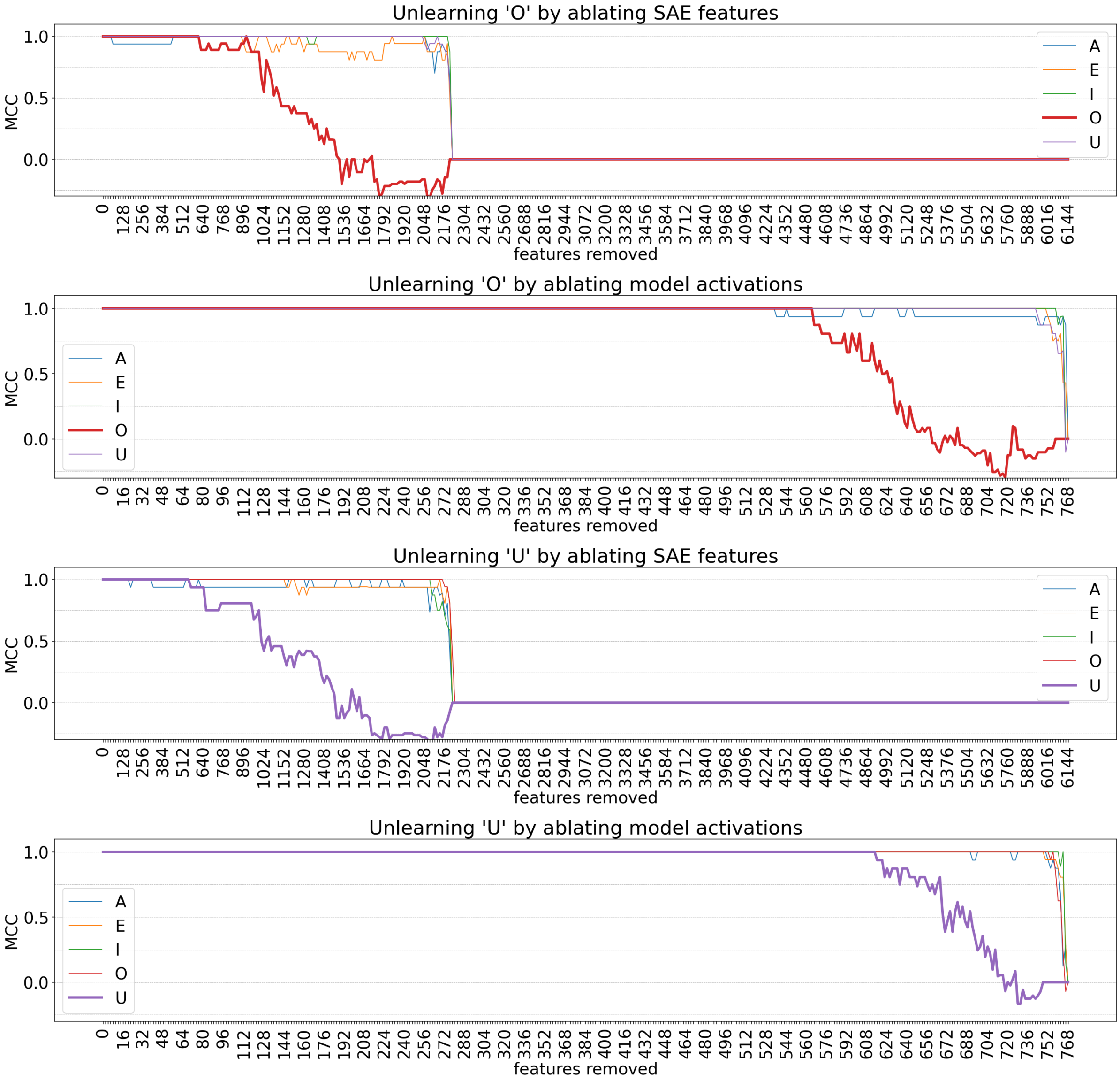


Figure 27: Unlearning plots for letters ’O’ and ’U’ at the last layer of HuBERT model, using `LogisticRegression` without regularization and `max_iter=10000`

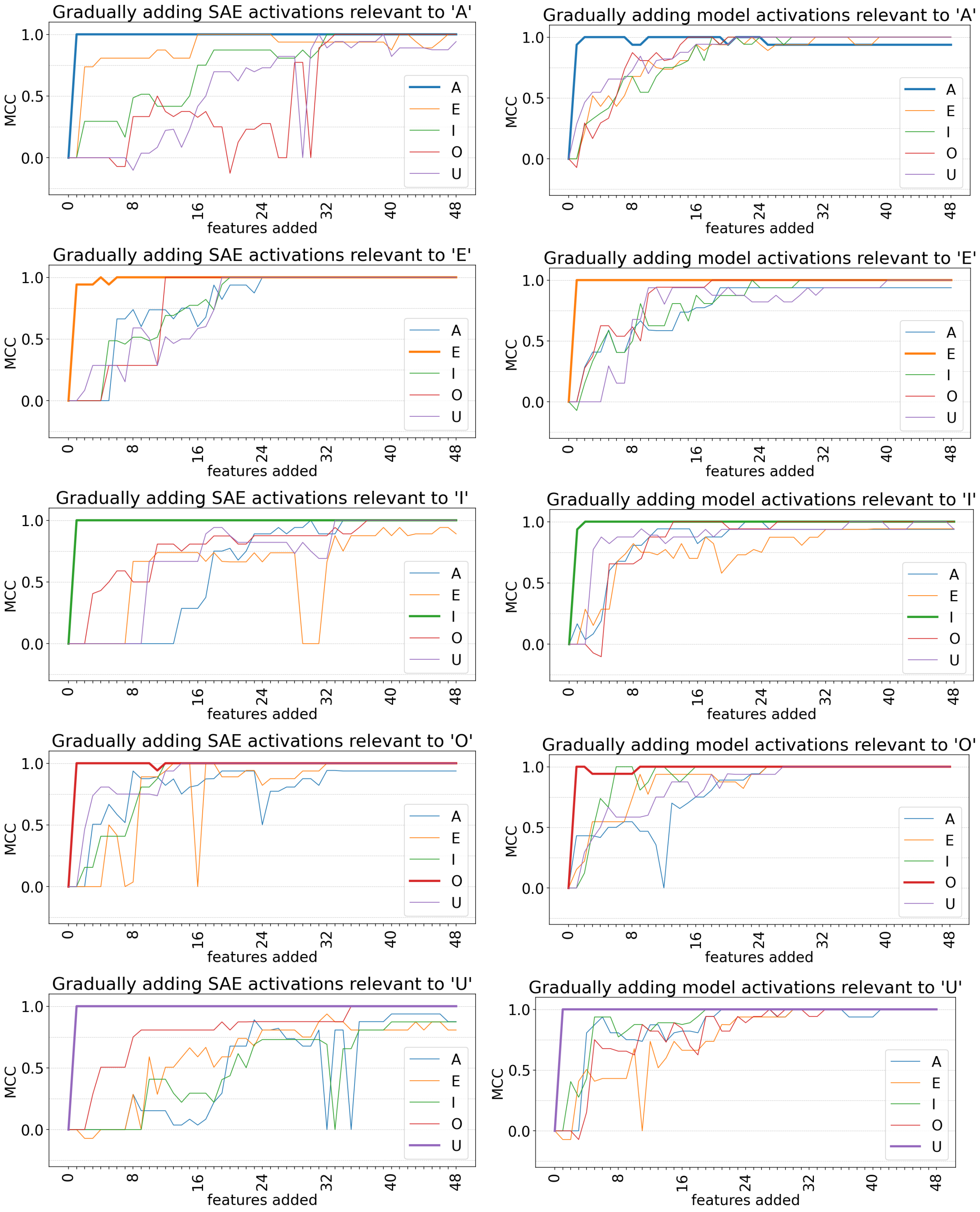


Figure 28: K-probe vowel classification at layer 9. The curves show classification accuracy as the most informative features are added sequentially. Only the first 49 features are displayed; beyond this point, accuracy approaches perfect accuracy for all vowels.