Difference in Task Performance on Sparse Speech Representations

Wenjie Peng, Chen Chen, Thomas Hain
School of Computer Science, The University of Sheffield, UK
{wpeng16,chen.chen2,t.hain}@sheffield.ac.uk

Abstract

Learning speech representations that are useful for a variety of downstream tasks has received considerable attention, due to the outstanding properties of Self-Supervised Learning (SSL) trained models. Despite advancements in modelling methods, understanding the difference in task performance on representations is limited. Mainly motivated by the no-free-lunch theorem and speech production, this work investigates changes in task performance in sparse speech representations, providing interpretability analysis under the Information Bottleneck (IB) framework. Autoencoders with varying sparsity levels were trained using three SSL features, and evaluated on six tasks of SUPERB: Speech Enhancement (SE), Speaker Identification (SID), Speech Emotion Recognition (SER), Phone Recognition (PR), Automatic Speech Recognition (ASR) and Slot Filling (SF). Experiments show that: 1) different tasks manifest different degrees of sensitivity to the sparsity levels; 2) the optimal sparsity level for task performance varies; 3) the choice of SSL features has a limited impact on most tasks but with an exception of PR; 4) overall PR and ASR require more preservation of relevant information about the labels, while SID and SER demand more compression of irrelevant information, where the input quality can shift this trade-off to some degree. These findings can contribute to the design of a universal sparse speech representation learner.

1 Introduction

The performance of machine learning models is usually highly dependent on the input representations (Bengio et al., 2013). Methods for learning useful representations have recently been extensively investigated in different domains (Devlin et al., 2019; Chen et al., 2020; Mohamed et al., 2022). Inspired by the success in the field of Natural Language Processing (NLP) (Devlin et al., 2019), one important topic in modelling of speech is the learning of representations that are useful for a wide range of downstream tasks using Self-Supervised Learning (SSL). SSL is a type of unsupervised learning that utilises supervision from parts of the input data itself. SSL models are typically optimised by solving a so-called pretext task (Mohamed et al., 2022). Such a pretext task can be reconstructing the original input (Van Den Oord et al., 2017; Huang et al., 2022), distinguishing positive from negative samples (Baevski et al., 2020) and predicting targets from an offline teacher model (Hsu et al., 2021; Chen et al., 2022) or with the self-distillation framework (Baevski et al., 2022; Liu et al., 2023). When evaluating SSL models on a wide range of downstream tasks, the learned self-supervised speech representations demonstrate robust performance compared to traditional features as model input (Yang et al., 2021).

Despite rapid progress in learning useful general-purpose speech representations with SSL, the understanding of the difference in task performance on the learned representations is limited. Difference in task performance, or task difference in short, in this work, refers to *how task performance differs with differently learned representations*. Therefore, knowledge of the task difference can contribute to improving the performance on all tasks, pursuing the goal of learning useful general-purpose speech representations. Motivated by the no-free-lunch theorem in statistical learning theory that one needs inductive bias to encode task-specific information (Shalev-Shwartz and Ben-David, 2014), this work investigates the task difference in the inductive bias of sparsity for representation learning. The motivation for choosing sparsity as the inductive bias is two-fold. First, according to (Bengio et al., 2013), "Good representations are expressive, meaning that a reasonably-sized learned representation can capture a huge number of possible input configurations." As a result, one way to improve expressiveness is to learn sparse repre-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 28278–28291

sentations (Bengio et al., 2013). Second, from a speech-oriented view, a speech signal is produced by slowly adjusting the gesture of a small number of articulators (Deng, 1999). Based on this, previous studies have suggested that sparsity can effectively induce distinct task-specific information (Hande Kabil and Bourlard, 2022; Deng et al., 2013; Sisman et al., 2018; Lu et al., 2022; Peng et al., 2022). However, a comprehensive study on the inductive bias of sparsity for useful general-purpose speech representations is underexplored. Furthermore, as most current speech SSL models have demonstrated superior performance with dense representations (Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022), their potential after sparsification requires empirical investigation. To fill the gaps, this work aims to investigate the task difference in sparse speech representations across a wide range of tasks.

To achieve the above objective, two questions are addressed: 1) *how do different tasks perform under the varying sparsity levels?* 2) *what can we learn if a certain sparsity level is necessary for optimal task performance?* To answer these questions, autoencoders with varying sparsity levels were trained using three commonly-used SSL features as input and evaluated on six tasks from SUPERB with a wide coverage of speech properties, including Speech Enhancement (SE), Speaker Identification (SID), Speech Emotion Recognition (SER), Phone Recognition (PR), Automatic Speech Recognition (ASR) and Slot Filling (SF). The selected six tasks cover five out of six domains from SUPERB (Yang et al., 2021), including generation, speaker, paralinguistics, recognition and semantics, evaluating the capability of useful general-purpose. Then, the principle of Information Bottleneck (IB) was adopted to interpret the relationship between tasks and their associated sparsity levels for optimal performance. The main contributions of this work are:

- A first comprehensive investigation of the inductive bias of sparsity for speech SSL models on general-purpose speech representation learning, suggesting a suitable sparsity level is necessary for optimal task performance. (Section 4.2).
- Interpretability analysis under the information bottleneck framework, highlighting the compression-preservation preference for SID, SER, PR and ASR, respectively (Section 4.3).

2 Related Work

2.1 Sparse Representation Learning

Motivation and related work on learning sparse representations mainly originates from findings in neuroscience, which suggest that the human brain adopts sparse coding mechanisms for efficiency (Olshausen and Field, 1997). In machine learning, learning sparse representations has been extensively studied using autoencoders (Makhzani and Frey, 2014; Bricken et al., 2023; Gao et al., 2025). Compared to their dense counterpart, Sparse Autoencoder (SAE) typically projects the raw input data into a higher-dimensional space. Without any constraints, the resulting overcomplete autoencoders may struggle to learn meaningful representations. The commonly used constraint is to induce sparsity by minimising a regularisation term such as the L_1 norm of the activations (Bricken et al., 2023) or placing a TopK module in the encoder part (Makhzani and Frey, 2014). Optimising the L_1 norm has the problem of suppressing the learned representation (Rajamanoharan et al., 2024), i.e. the magnitude of the sparse representation tends to shrink towards zero. In contrast, k-sparse autoencoders can mitigate this issue by directly controlling the number of zeros using the TopK operation (Gao et al., 2025). A recent study further enhanced the k-sparse autoencoders (Gao et al., 2025), as covered in Section 3.1, which will be used as the model backbone for this work. In contrast to the rising interest in extracting interpretable features from the activations of Large Language Models (LLMs) on text (Gao et al., 2025; Bricken et al., 2023; Lieberum et al., 2024; Huben et al., 2024), this work aims to reveal the relationship between task performance differences in sparse speech representations.

2.2 The Principle of Information Bottleneck

IB provides an information-theoretical view for interpreting the performance of supervised trained deep neural networks (Tishby et al., 2000; Tishby and Zaslavsky, 2015): their performance relies on the trade-off between compression of the data and preservation of the relevant information about the target labels for prediction. The IB principle quantifies data compression by the Mutual Information (MI) $I(x; z)$ between the input data x and the intermediate representations z from deep neural networks, while the preservation of relevant information is quantified by the MI $I(z; y)$ between

z and target labels y. The optimal performance can be achieved when $I(x;z)$ is minimised while $I(z;y)$ is maximised, i.e. z has extracted the minimal sufficient information from x about y. Since its establishment, IB has emerged as a foundational framework for interpreting the performance of supervised deep learning models. While IB has been explored in various speech tasks with a fine-tuned balance between compression and preservation (Zhang et al., 2021; Stafylakis et al., 2024; Si et al., 2021), it remains unclear how different tasks differ in managing this trade-off. The principle of IB will be adopted in the analysis of task-specific differences in sparse speech representations.

3 Sparse Representations for Task-specific Inductive Bias

This section introduces the methods used throughout this work. Section 3.1 presents the background on k-sparse autoencoders (Makhzani and Frey, 2014; Gao et al., 2025), which is used for sparse speech representation learning. Subsequently, Section 3.2 defines an information-theoretical measure to quantify the preservation of relevant information about target labels, connecting to the IB framework.

3.1 k-Sparse Autoencoders

Mathematically, the sparsity level can be measured by counting the number of non-zero elements, i.e. L_0 norm (Makhzani and Frey, 2014; Gao et al., 2025). The k-sparse autoencoder (Makhzani and Frey, 2014) imposes sparsity by retaining the k largest activations (TopK operation), through constraining the L_0 norm of the latent representation vectors to exactly k non-zero elements. The larger the k, the higher the sparsity level (Makhzani and Frey, 2014). During the forward pass, the TopK operator keeps the k largest hidden units active while zeroing the rest. During the backward pass, only the active hidden units receive gradients. Let x denote a speech frame vector where $x \in \mathbb{R}^d$ and z denotes its associated latent representation where $z \in \mathbb{R}^h$, $d < h$, d and h are the dimensionality for x and z, respectively. Following (Gao et al., 2025), the encoding and decoding processes are defined as:

$$z = \text{TopK}(W_{enc}(x - b_{pre}) + b_{enc}), \quad (1)$$

$$\hat{x} = W_{dec}z + b_{pre}, \quad (2)$$

where W_{enc} and W_{dec} are the weights for encoder and decoder respectively, and b_{pre} and b_{enc} are the pre-encoder and encoder bias (Bricken et al., 2023). The k-sparse autoencoder is optimised using the MSE reconstruction loss between the reconstructed input $\hat{x}$ and x: $L_{mse} = ||x - \hat{x}||_2^2$. Note that the averaging over N data samples for L_{mse} is omitted for simplicity.

Optimising SAE in practice is difficult, as it is prone to the problem of so-called dead latent representations or dead latents for short, i.e. some representation dimensions remain inactive when the input speech frames are sufficiently large. To mitigate the issue of dead latents, two techniques have been proposed for the k-sparse autoencoders (Gao et al., 2025). The first is to tie the weights of the decoder to the transpose of those in the encoder for initialisation. The second is to use an auxiliary loss L_{aux} to minimise the reconstruction error using the top-k$_{aux}$ dead latents z_{aux}, together with the final loss L, which are defined as:

$$L_{aux} = ||(x - \hat{x}) - W_{dec}z_{aux}||_2^2, \quad (3)$$

$$L = L_{mse} + \lambda L_{aux}, \quad (4)$$

where λ is the weight for the auxiliary loss. Note the sum over N samples in Equation 3 is also omitted. In practice, determining the dead latents z_{aux} usually requires a threshold τ. Instead of using a fixed value as in Gao et al. (2025), τ is defined as the ratio of being inactive within a batch in this work. For a batch with N frames and a latent remained zero from n frames, it was considered dead if n/N was larger than τ. Defining τ in this way has better control for batch updates consisting of variable-length sequences.

3.2 Measuring Relevant Information

In a k-sparse representation z, each latent z_i has two states, being active or not. Therefore, the state for z_i can be parametrised with a Bernoulli distribution with the probability of being active as $p(z_i)$ and being inactive as $1 - p(z_i)$. Given N data samples, the empirical estimation of $\hat{p}(z_i)$ is defined as:

$$\hat{p}(z_i) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}_{z_i^j > 0}, \quad (5)$$

where $\mathbf{1}_{z_i^j>0}$ is the indicator function that takes the value one when z_i is activated given the j-th speech frame. Since the sparsity levels are induced in an unsupervised way, the variation of latents

being active should reflect the property of the data. When $\hat{p}(z_1), ..., \hat{p}(z_h)$ is i.i.d.[1], such variation can be quantified as below (Cover and Thomas, 2006):

$$H(z) = -\sum_{i=1}^{h}(\hat{p}(z_i)\log(\hat{p}(z_i)) + (1-\hat{p}(z_i))(\log(1-\hat{p}(z_i)))), \quad (6)$$

where $H(z)$ is the entropy of the latents. In terms of speech, it contains various properties, such as speaker identity, emotional state, phone and word, etc. Let y be the categorical random variable that represents a speech property that takes M classes (e.g. y can be the labels for speaker identity, emotion state, phone or word, respectively). The variation of latents being active given the label y, i.e. $H(z \mid y)$, can be quantified by:

$$H(z \mid y) = \sum_{j=1}^{M}\hat{p}(y_j)H(z \mid y_j), \quad (7)$$

$$H(z \mid y_j) = -\sum_{i=1}^{h}(\hat{p}(z_i|y_j)\log\hat{p}(z_i \mid y_j) + (1-\hat{p}(z_i \mid y_j))\log(1-\hat{p}(z_i \mid y_j))), \quad (8)$$

where $H(z \mid y_j)$ is the conditional entropy of the latents z given label y_j, $\hat{p}(y_j) = \frac{N^j}{N}$ is the empirical estimation of the probability for the label being the j-th class, and N^j is the number of frames associated with y_j. $\hat{p}(z_i|y_j) = \frac{N_i^j}{N^j}$ represents the probability of z_i being active given y_j, N_i^j denotes the number of frames for z_i being active given y_j. Since the downstream tasks usually require access to the information about y, it is important to quantify the shared information between z and y, which is defined as below:

$$I(z;y) = H(z) - H(z \mid y), \quad (9)$$

where $I(z;y)$ is the MI between z and y. From the view of IB, higher $I(z;y)$ means the representations z preserve more information about the label y, which is usually helpful for a supervised model to predict y. However, the prediction performance not only relies on the prediction power but also on the compression of data. The process of sparsification can be seen as a form of compression as it transforms the data into low-rank subspaces. Concretely, the measure of sparsity and entropy is closely related, where a higher sparsity usually indicates a lower entropy (Pastor et al., 2013). Furthermore, entropy can be used as a regularisation term, which has been demonstrated to impose a sparsity solution through optimisation (Huang and Tran, 2019). Consequently, sparse coding has been widely applied in the field of compressed sensing (Donoho, 2006). The level of compression for the sparse speech representations can be controlled by the magnitude of k as it effectively adjusts the L_0 norm. Empirically, the smaller k, the higher level of data compression. According to IB, the optimal performance always makes a trade-off between data compression and the preservation of the relevant information about labels. To investigate how tasks differ in making such trade-offs, four tasks, including SID, SER, PR and ASR were adopted, interpreting the optimal sparsity level for tasks.

[1]Supported by empirical evidence that $I(z_i;z_j) \approx 0.0$ for $i \neq j, i, j \in \{1, 2, \ldots, h\}$

4 Experiments

4.1 Experimental Setup

For training of the SAE, the train-clean-360 subset with 360h clean data from the Librispeech (Panayotov et al., 2015), a read English corpus, was adopted. The raw waveforms were first converted into 768-dimensional dense representations using a pretrained SSL model. In this work, the SSL model was used as a fixed feature extractor, i.e. the model remained unchanged throughout training and evaluation of the SAE. Since the speech SSL models based on contrastive learning and predictive coding perform differently as suggested by the information analysis (Pasad et al., 2021), to investigate the impact of the choice for SSL models, three commonly used SSL models of WavLM [2], HuBERT [3] and wav2vec 2.0[4] base were adopted to train SAE. The experimental results and analysis throughout this section are based on the WavLM base features for SAE training, leaving that from the HuBERT and wav2vec 2.0 base in the Appendix A.

SAEs with different sparsity levels were trained by sweeping the dimensionality $\in \{1536, 2304, 3072, 3840, 4608\}$ (2-6 times the dimensionality of 768 from WavLM base features) and sparsity level constraint $k \in$

[2]https://huggingface.co/s3prl/converted_ckpts/resolve/main/wavlm_base.pt

[3]https://huggingface.co/s3prl/converted_ckpts/resolve/main/hubert_base_ls960.pt

[4]https://huggingface.co/s3prl/converted_ckpts/resolve/main/wav2vec_small.pt

Dimensionality	k	SE		SID	SER	PR	ch-ASR	wp-ASR	SF	
		STOI ↑	PESQ ↑	Acc ↑	Acc ↑	PER ↓	WER ↓	WER ↓	F1 ↑	CER ↓
1536	32	84.65	1.98	**50.59**	**67.96**	7.71	6.54	6.72	88.25	26.93
	64	84.77	1.98	50.17	67.57	6.55	6.17	6.25	88.62	**26.37**
	128	84.62	1.98	50.10	67.32	5.98	5.88	5.99	**88.61**	26.43
	256	**84.90**	**1.98**	50.17	67.86	**5.65**	**5.69**	**5.95**	88.37	26.52
2304	32	84.60	1.97	**55.79**	67.82	7.13	6.48	6.65	88.80	26.11
	64	84.84	1.98	50.98	**68.29**	5.84	6.08	6.14	88.62	26.74
	128	84.84	1.98	52.53	67.83	5.36	5.90	**5.86**	**89.01**	**25.46**
	256	**84.87**	**1.99**	51.48	67.62	**5.19**	**5.74**	5.91	88.49	26.69
3072	32	84.37	1.96	**57.56**	**69.64**	6.57	6.51	6.94	87.76	27.63
	64	84.56	1.97	52.43	68.27	5.53	5.90	6.24	88.71	26.10
	128	**84.70**	**1.99**	53.20	68.20	5.03	5.80	6.08	88.41	**26.02**
	256	84.69	1.99	52.94	68.46	**4.99**	**5.73**	**6.01**	**88.87**	26.38
3840	32	84.34	1.97	**57.43**	**69.38**	6.65	6.69	6.97	88.29	27.89
	64	84.55	**1.99**	53.94	68.85	5.21	5.96	6.12	88.47	26.72
	128	**84.81**	1.98	53.52	68.56	4.88	5.83	6.07	**89.04**	**25.73**
	256	84.74	1.99	53.06	68.85	**4.86**	**5.81**	**5.95**	88.81	25.90
4608	32	84.43	1.96	**58.14**	68.83	6.50	6.50	6.83	88.02	27.19
	64	84.55	1.99	55.42	**69.31**	5.19	5.94	5.90	88.84	25.95
	128	**84.67**	**1.99**	53.87	68.44	4.77	5.83	5.82	**89.10**	**25.56**
	256	84.61	1.99	53.33	68.50	**4.67**	**5.76**	**5.71**	88.61	28.28

Table 1: Evaluation results of sparse speech representations on the dev set from the six tasks, with varying sparsity levels based on WavLM features as input for SAE.

$\{32, 64, 128, 256\}$. The number of parameters for SAE across different dimensionality h is 2.4M, 3.5M, 4.7M, 5.9M and 7.1M, respectively. As will be shown in Section 4.2 that the pattern across the existing h is quite similar, so the dimensionality is not further increased in the experiments. The assignment of k follows Gao et al. (2025), where further increasing k degrades the quality of the representations. The Adam optimiser (Kingma and Ba, 2015) was adopted with an initial learning rate of 1e-3, which was annealed by a factor of 0.8 once the MSE loss was not further improved on the dev-clean subset. The SAE were trained using 4 Nvidia 3090 GPUs with a total batch size of 512. The training of each SAE will finish after around 4h with 100 epochs, and the models that performed best on the dev-clean subset will be selected for evaluation. The implementations of SAE in this work mainly refer to [5]. Following (Gao et al., 2025), λ and top-k_{aux} was set as $1/32$ and 384 respectively. The optimal value of τ was set as 0.9999, based on the evaluation performance on the task of PR.

[5] https://github.com/openai/sparse_autoencoder (v0.1)

4.2 Varying Sparsity Levels

Evaluation protocol. To investigate how different tasks perform on different sparsity levels, six tasks from SUPERB (Yang et al., 2021) were used. The setup for each task follows the original implementation [6], where SAE was kept frozen during evaluation. The datasets used for the six tasks are in English, with one exception from SID, where the data were collected 'in the wild' (Nagrani et al., 2017). For the ASR task, both Char based ASR (ch-ASR) and Wordpiece based ASR (wp-ASR) systems were evaluated to investigate the impact of the number of acoustic units. The number of acoustic units is 32 and 300 for ch-ASR and wp-ASR, respectively. The evaluation metrics for each task follow (Yang et al., 2021), and the performance on their associated dev set is shown in Table 1.

Optimal sparsity level selection. Results in Table 1 show that the optimal sparsity level for different tasks varies significantly. For both PR and ch-ASR, when the dimensionality is given, increasing k will bring better performance, and the optimal performance always comes from the combination

[6] https://github.com/s3prl/s3prl (v0.4.14)

of the largest dimensionality and k, indicating the importance of both higher representation capacity and higher sparsity level. The pattern for wp-ASR is similar to ch-ASR but with one exception when the dimensionality is 2304. This suggests that the number of acoustic units has a limited impact on selecting the sparsity level for optimal performance on the ASR task. For SE, the optimal performance per dimensionality usually comes from a relatively large k, indicating the preference for a higher sparsity level, i.e. a larger k. For SF, when the dimensionality is given, the optimal performance usually comes from an intermediate k, and the metrics of F1 and CER show a slight difference in the selection of the optimal k. While the best performance on F1 requires a higher dimensionality, CER is lowest with an intermediate value, indicating that the representation capacity impacts the two metrics differently. Compared to other tasks, the optimal performance for SID and SER requires a relatively low sparsity level. For SID, the optimal performance is always observed with the smallest k for a given dimensionality, and the best performance is obtained when the dimensionality is the highest, indicating the importance of large representation capacity and a low sparsity level. For SER, the optimal performance comes from a relatively small k when the dimensionality is given, and it achieves the best performance with an intermediate dimensionality and the smallest k. Overall, PR and ch-ASR require the highest sparsity level per dimensionality h, followed by wp-ASR and SE, which perform effectively with relatively high sparsity levels. While SF benefits from intermediate sparsity levels, SID and SER perform best with low sparsity levels. The overall tendency is consistent with the SAE trained with HuBERT features but with slight differences for wav2vec 2.0 as shown in Appendix A.1. For SE and SF that are evaluated with two metrics, the correlation between them is strong enough, so the performance is consistent across different sparsity levels as suggested by a correlation analysis in Appendix A.2.

Relative improvement. To better examine the performance range for the individual tasks, the final layer features from the WavLM base model were evaluated directly on the above tasks to calculate the relative improvement, which is illustrated in Figure 1. From Figure 1 it can be seen that different tasks manifest different degrees of sensitivity to the varying sparsity levels. Compared to SE, which appears to be most resilient to different sparsity levels, the other tasks show visible variation, indicating the importance of a suitable sparsity level for most tasks. Among these tasks, PR exhibits the largest performance variation across different sparsity levels, followed by wp-ASR and ch-ASR. While the relative improvement for SID and SER varies at an intermediate level, SF shows a relatively small change. For SF, the varying sparsity levels have more influence on the metric of CER rather than F1. Overall, it is clear from Figure 1 that no single sparsity level can dominate all the tasks, suggesting the importance of fine-tuning suitable sparsity levels for individual tasks. More precisely, the best performance regarding the nine metrics in Figure 1 is 0.32%, 1.22%, 12.72%, 4.24%, 10.02%, 1.90%, 3.55%, 0.79%, 5.53%, respectively, all leading to positive relative improvement.

Figure 1: Relative improvement of sparse speech representations on the dev set for each task. h is the dimensionality for the latents, while k denotes the number of their non-zero elements. WavLM features are used as input for SAE.

Comparison of different speech SSL models. SAE trained with the HuBERT and wav2vec 2.0 features are shown in Appendix A.3. In contrast to WavLM, HuBERT and wav2vec 2.0 features have degraded performance on some tasks. For HuBERT, sparse representations can improve on most tasks and are on par with the baseline for SF but slightly worse performance on PR, where the best relative improvement is -0.02%, -0.17% and -3.39%, respectively. The inferior performance on PR may be due to the fact that the correlation with phone labels decreases in the last two layers (Pasad et al., 2023). Similarly, wav2vec 2.0-based sparse representation can improve on most tasks, but with exceptions of SER and PR, where the best relative improvement is -1.36% and -16.61%, respectively.

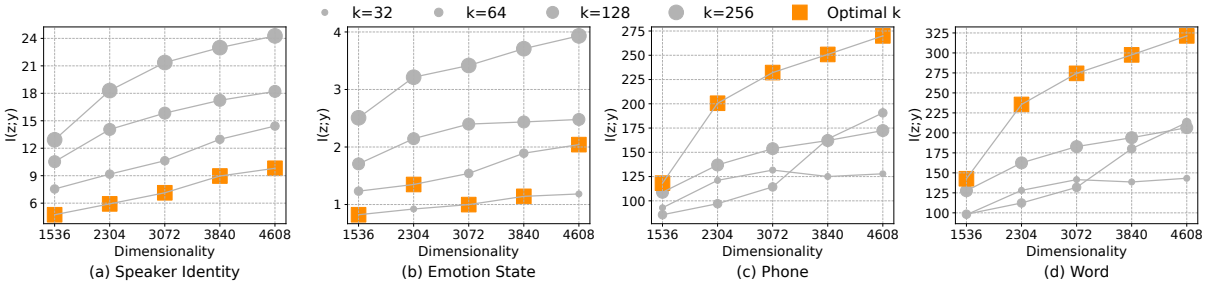

Figure 2: Mutual information in bits between latents and task-specific labels for speaker identity, emotion state, phone and word, respectively. WavLM features were used as input for SAE.

As suggested by (Pasad et al., 2021, 2023), WavLM has the best correlation with phone labels on top layers, while the significantly degraded correlation for wav2vec 2.0, which may account for the above performance on PR. This comparative study suggests that despite the commonly-used SSL models can provide improvement for sparse speech representations on most tasks, the practical choice can have an impact on some tasks, especially for PR.

4.3 Interpretation of Optimal Sparsity Levels

Setup. To interpret the relationship between tasks and their associated sparsity levels for optimal performance, four tasks were adopted for this purpose, including SID, SER, PR and ASR. Since wp-ASR exhibits a similar pattern to ch-ASR, ASR refers to the ch-ASR in this subsection. For SID, SER, PR and ASR, the MI defined in Section 3.2 is calculated on the dev set, where y is represented by the labels of speaker identity, emotion state, phone and word, respectively. The associated numbers of the classes on the dev set are 1251, 4, 39 and 2969, respectively. For both phone and word labels, silence was removed, and the segments were excluded if their associated word had fewer than 50 samples. The dev set for each task was adopted for the MI calculation, and Figure 2 illustrates its relationship with varying dimensionality h and k.

Interpretation under the framework of IB. It can be seen from Figure 2 that when the dimensionality is given, increasing k will always improve the MI between the representations and labels for all four tasks. This suggests that a larger k is helpful to preserve the relevant information about labels. For the tasks of PR and ASR, the optimal performance is consistently obtained when k is the largest, indicating the requirement for more preservation of relevant information about the labels. In contrast, both SID and SER obtain the optimal performance when k is relatively small, indicating the requirement for less preservation of the relevant information about labels. Since a smaller k induces higher sparsity, it brings a higher level of data compression. From the view of IB, the optimal performance for a supervised task relies on the trade-off between data compression and the preservation of relevant information about the labels. Therefore, compared to PR and ASR, SID and SER require more compression of data while less preservation of relevant information about the target labels for optimal performance. This observation also holds for the SAE trained with HuBERT-based features, but is different for wav2vec 2.0 on SER as shown in Appendix A.4. Similar to Section 4.2, HuBERT-based features exhibit similar behaviours like WavLM but distinguish from wav2vec 2.0. Note that this conclusion is based on the evaluation protocol from SUPERB (Yang et al., 2021), where only PR and ASR share the same corpus, while SID and SER are evaluated on distinct corpora, all of which are commonly used for evaluating the tasks.

The impact of input quality. Further experiments were conducted to elaborate on the reason why some tasks require distinct sparsity levels, which is used as complementary evidence to the above IB interpretation. The experimental results suggest that: 1) although the optimal sparsity level seems to be more correlated with individual tasks, the input quality can shift the optimal k to some degree; 2) SAEs can still improve the performance in some cases even if the information is noisy. Please refer to Appendix B for more details.

4.4 Ablation Study

To determine the optimal value of τ, i.e. the threshold to spot the dead latents, SAE with dimensionality of 1536 and k of 32 was trained on the train-clean-100 subset of Librispeech (Panay-

otov et al., 2015) and evaluated on the PR task. WavLM features were used to provide the input for SAE. τ was swept across values $\in \{0.9, 0.99, 0.999, 0.9999, 0.99999, 1.0\}$ and the performance on the dev-clean subset is shown in Table 2.

τ	PER
0.9	12.22
0.99	11.49
0.999	10.78
0.9999	**9.98**
0.99999	10.05
1.0	10.33

Table 2: Phone error rate for the phone recognition task using sparse speech representations on the dev-clean subset with different threshold τ.

It can be seen in Table 2 that the performance will increase first, reach an optimal point when $\tau = 0.9999$, and then decrease when τ is kept increasing. Therefore, τ is set as 0.9999 throughout this work.

5 Discussions and Conclusions

Self-supervised speech representation learning aims to obtain useful general-purpose representations for a wide range of downstream tasks. Since different tasks usually differ in the task-specific information, understanding the task performance difference is important to learn better representations and thus improve performance on all tasks. Mainly motivated by the no-free-lunch theorem and speech production, this work investigated the task difference in sparse speech representations for encoding task-specific information. A first comprehensive study on the inductive bias of sparsity for speech SSL models to learn useful general-purpose speech representations is presented, with three commonly-used SSL features for SAE training, each SAE is experimented with twenty configurations and each configuration is evaluated on six tasks of SUPERB. The experimental results reveal the relationship between task performance and the sparsity levels in the speech representations. With the interpretability analysis under the IB, the sparsity level for optimal performance suggest that overall SID and SER require more on data compression, while PR and ASR demand more on the preservation of relavent information about labels. Further empirical evidence suggests that the input quality has a slight control on this trade-off. These findings have the potential to contribute to the design of universal speech representation learning with the inductive bias of sparsity.

Limitations

This work has several limitations. First, the SSL features to train SAE are from the base model, where future work may explore their large counterparts as input. Second, the training and evaluation of SAE are mainly based on English data, where the analysis in the multilingual scenario is beyond the scope of this work. Third, the measure of MI only considers the status of latents being active or not, i.e. modelled with the Bernoulli distribution, while ignoring their magnitudes. Future work may explore a more complex model to quantify the distribution of the k-sparse representations.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions to improve this work. This work was conducted at the VoiceBase/LivePerson Centre of Speech and Language Technology at the University of Sheffield, which is supported by LivePerson, Inc.

References

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proceedings of the 39th International Conference on Machine Learning*, pages 1298–1312. PMLR.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *In Proceedings of the 34th Advances in Neural Information Processing Systems*, 33:12449–12460.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning.

Transformer Circuits Thread. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA.

Jun Deng, Zixing Zhang, Erik Marchi, and Björn Schuller. 2013. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 511–516. IEEE.

Li Deng. 1999. Computational models for speech production. In *Computational models of speech pattern processing*, pages 199–213. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

D.L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

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 *Proceedings of the 13th International Conference on Learning Representations*.

Selen Hande Kabil and Herve Bourlard. 2022. From undercomplete to sparse overcomplete autoencoders to improve lf-mmi based speech recognition. In *Interspeech 2022*, pages 1061–1065.

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.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. 2022. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720.

Shuai Huang and Trac D. Tran. 2019. Sparse signal recovery via generalized entropy functions minimization. *IEEE Transactions on Signal Processing*, 67(5):1322–1337.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the 12th International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimisation. *International Conference on Learning Representations*.

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.

Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and Jim Glass. 2023. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. *Advances in Neural Information Processing Systems*, 36:58346–58362.

Yizhou Lu, Mingkun Huang, Xinghua Qu, Pengfei Wei, and Zejun Ma. 2022. Language adaptive cross-lingual speech representation learning with sparse sharing sub-networks. In *Proceedings of the 47th International Conference on Acoustics, Speech and Signal Processing*, pages 6882–6886. IEEE.

Alireza Makhzani and Brendan Frey. 2014. K-sparse autoencoders. *International Conference on Learning Representations*.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, and 1 others. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: A large-scale speaker identification dataset. In *Interspeech 2017*, pages 2616–2620.

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *Proceedings of the 40th International Conference on Acoustics, Speech and Signal Processing*, pages 5206–5210.

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

Ankita Pasad, Bowen Shi, and Karen Livescu. 2023. Comparative layer-wise analysis of self-supervised speech models. In *In Proceedings of the 48th International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE.

Giancarlo Pastor, Inmaculada Mora-Jiménez, Riku Jäntti, and Antonio J. Caamaño. 2013. Sparsity-based criteria for entropy measures. In *International Symposium on Wireless Communication Systems*.

Junyi Peng, Rongzhi Gu, Ladislav Mošner, Oldrich Plchot, Lukas Burget, and Jan Černocký. 2022. Learnable sparse filterbank for speaker verification. In *Interspeech 2022*, pages 5110–5114.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Shijing Si, Jianzong Wang, Huiming Sun, Jianhan Wu, Chuanyao Zhang, Xiaoyang Qu, Ning Cheng, Lei Chen, and Jing Xiao. 2021. Variational Information Bottleneck for Effective Low-Resource Audio Classification. In *Interspeech 2021*, pages 591–595.

Berrak Sisman, Mingyang Zhang, and Haizhou Li. 2018. A voice conversion framework with tandem feature sparse representation and speaker-adapted wavenet vocoder. In *Interspeech 2018*, pages 1978–1982.

Themos Stafylakis, Anna Silnova, Johan Rohdin, Oldřich Plchot, and Lukáš Burget. 2024. Challenging margin-based speaker embedding extractors by using the variational information bottleneck. In *Interspeech 2024*, pages 3220–3224.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, pages 1–5. IEEE.

Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee. 2021. Superb: Speech processing universal performance benchmark. In *Interspeech 2021*, pages 1194–1198.

Guangyan Zhang, Ying Qin, Daxin Tan, and Tan Lee. 2021. Applying the Information Bottleneck Principle to Prosodic Representation Learning. In *Interspeech 2021*, pages 3156–3160.

A SAE Trained with HuBERT and wav2vec 2.0 Features

A.1 Evaluation Results

The evaluation results of SE, SID, SER, PR, ch-ASR, wp-ASR and SF with varying sparsity levels of speech representations are shown in Table 3 and Table 4 with respect to HuBERT and wav2vec 2.0 features as input.

Dimensionality	k	SE		SID	SER	PR	ch-ASR	wp-ASR	SF	
		STOI ↑	PESQ ↑	Acc ↑	Acc ↑	PER ↓	WER ↓	WER ↓	F1 ↑	CER ↓
1536	32	84.70	1.97	**69.67**	67.67	12.49	8.50	8.31	83.64	33.65
	64	85.06	1.99	65.51	**67.74**	10.12	7.47	**8.13**	85.83	31.89
	128	85.17	2.01	66.45	67.40	8.49	6.97	8.14	86.77	30.14
	256	**85.26**	**2.02**	65.66	67.07	**7.64**	**6.74**	8.34	**86.94**	**29.93**
2304	32	84.66	1.98	**71.52**	**68.49**	11.10	8.28	8.46	84.15	34.13
	64	84.93	2.00	68.60	68.01	9.14	7.19	7.23	85.53	31.41
	128	85.12	**2.01**	67.73	67.97	7.60	6.67	**7.15**	86.35	**29.61**
	256	**85.17**	2.01	66.38	68.03	**6.98**	**6.52**	7.23	**86.41**	30.91
3072	32	84.49	1.96	71.93	68.20	11.01	8.25	7.14	84.32	33.02
	64	84.94	1.98	**73.62**	**69.48**	8.49	7.44	7.46	84.78	32.63
	128	85.10	**2.00**	67.71	68.34	7.08	6.64	6.74	**86.58**	**29.91**
	256	**85.20**	2.00	68.12	67.73	**6.70**	**6.34**	**6.54**	86.18	30.62
3840	32	84.62	1.96	71.91	68.10	10.98	8.28	6.51	84.89	33.44
	64	84.73	1.99	**75.04**	**69.23**	7.97	7.10	**6.50**	85.54	31.09
	128	85.06	2.00	68.61	68.66	6.82	**6.40**	6.51	**86.09**	**30.60**
	256	**85.23**	**2.01**	67.84	67.79	**6.53**	6.42	6.76	85.87	31.36
4608	32	84.60	1.97	70.83	68.29	11.42	8.37	6.51	83.82	33.29
	64	84.89	1.98	**74.62**	**69.02**	8.39	7.33	6.45	85.38	31.10
	128	85.02	2.00	69.21	68.52	6.71	6.38	**6.41**	**86.68**	**30.24**
	256	**85.10**	**2.01**	68.70	68.66	**6.40**	**6.33**	6.53	85.76	30.44

Table 3: Evaluation results of sparse speech representations on the dev set from the six tasks, with varying sparsity levels based on HuBERT features as input for SAE.

A.2 Correlation Analysis

To investigate the impact of sparsity levels on different metrics for a given task, the Pearson Correlation Coefficient (PCC) and the associated p-value were calculated using paired metrics of SE and SF for SAEs trained on wav2vec2.0, HuBERT, and WavLM features. The results are shown in Table 5.

A.3 Relative Improvement

The relative improvement of SAE over the baseline of HuBERT and wav2vec 2.0 features for individual tasks is illustrated in Figure 3 and Figure 4, respectively.

For Figure 3, the best relative improvement across the nine metrics is 0.07%, 1.07%, 13.23%, 2.81%, -3.39%, 3.41%, 0.47%, -0.02%, -0.17%, respectively. For Figure 4, the best relative improvement across metrics is 0.09%, 0.57%, 7.20%, -1.36%, -16.61%, 1.15%, 3.28%, 1.67%, 6.26%, respectively.

A.4 Measuring Task-specific Information

The MI between sparse speech representations and target labels is illustrated in Figure 5 and Figure 6, where the former is based on HuBERT features as input while the latter is on wav2vec 2.0.

Dimensionality	k	SE		SID	SER	PR	ch-ASR	wp-ASR	SF	
		STOI ↑	PESQ ↑	Acc ↑	Acc ↑	PER ↓	WER ↓	WER ↓	F1 ↑	CER ↓
1536	32	84.63	1.95	**47.91**	**64.97**	46.70	13.21	12.93	78.61	42.36
	64	85.00	1.97	45.76	64.94	42.50	11.72	11.42	77.57	43.57
	128	**85.24**	1.99	45.87	64.48	38.46	10.44	10.20	78.32	41.43
	256	**85.24**	**2.01**	46.74	64.60	**35.10**	**9.46**	**9.32**	**79.58**	**39.41**
2304	32	84.63	1.94	48.12	64.17	46.08	13.68	13.37	79.12	41.75
	64	84.94	1.97	48.96	64.81	39.91	11.44	11.59	78.47	42.66
	128	**85.17**	1.98	48.19	64.89	37.35	10.61	10.17	**79.45**	**40.16**
	256	**85.17**	**1.99**	**49.20**	**65.45**	**34.32**	**9.62**	**9.60**	78.60	42.39
3072	32	84.52	1.94	49.64	63.99	44.35	13.08	14.47	78.92	41.71
	64	84.66	1.96	51.22	64.09	40.68	11.72	11.41	79.03	40.98
	128	85.09	1.98	49.81	64.90	35.96	10.46	10.50	78.87	41.14
	256	**85.28**	**2.00**	**49.96**	**65.15**	**34.17**	**9.67**	**9.17**	**79.82**	**40.06**
3840	32	84.59	1.94	47.58	64.73	45.91	14.00	13.60	79.40	40.10
	64	84.73	1.96	**51.51**	64.94	37.32	11.26	11.40	79.07	41.33
	128	85.18	1.98	50.52	64.96	36.02	10.22	10.16	78.83	41.24
	256	**85.22**	**1.99**	50.23	**64.98**	**34.18**	**9.68**	**9.47**	**79.56**	**39.60**
4608	32	84.45	1.94	47.25	63.81	46.25	14.56	13.92	77.26	42.87
	64	84.66	1.96	**52.46**	64.83	38.07	11.69	11.21	79.17	**39.88**
	128	85.02	1.99	50.94	65.14	35.25	10.96	10.25	78.89	41.70
	256	**85.40**	**2.00**	50.59	**65.16**	**33.48**	**9.54**	**9.07**	**79.48**	40.46

Table 4: Evaluation results of sparse speech representations on the dev set from the six tasks, with varying sparsity levels based on wav2vec 2.0 features as input for SAE.

	SE	SF
wav2vec 2.0	0.95 (p=0.000000)	-0.87 (p=0.000001)
HuBERT	0.93 (p=0.000000)	-0.93 (p=0.000000)
WavLM	0.71 (p=0.000494)	-0.83 (p=0.000007)

Table 5: Pearson correlation coefficients and p-values for the paired metrics of SE and SF with SAEs trained on different SSL features.

B Impact of the Input Quality

In Section 4.3 it suggests that different tasks require different sparsity levels for optimal performance. As the SAEs above were trained only on the last layer of speech SSL models, the input quality may be an important factor for the performance. For example, compared to wav2vec 2.0, the last layer of WavLM is more correlated with linguistic features (Pasad et al., 2023) so it is likely to improve the performance for tasks like PR and ASR. Furthermore the first layer preserves more information with respect to data compared to the last layer of speech SSL models, according to the data-processing inequalities (Cover and Thomas, 2006). To investigate the impact of input quality on the selection of optimal sparsity levels, SAEs were also trained on the first layer to provide some empirical evidence.

Due to time limitations, only WavLM and wav2vec 2.0 features were adopted to train SAEs, where the hidden dimensionality is set as 4608, with k sweeping from {32,64,128,256}, i.e. four SAEs for each speech SSL model. As for the evaluation, similar to Figure 1 and Figure 4, the following relative improvement (in %) is calculated: for PR it was for both SSL models, while SID was only for wav2vec 2.0. The results are shown in Table 6.

Overall the results in Table 6 suggest that the input quality has a slight impact on the compression-preservation trade-off and SAEs are useful for downstream tasks. By comparing exp 3 and 4, it is

Figure 3: Relative improvement of sparse speech representations on the dev set for each task. h is the dimensionality for the latents, while k denotes the number of their non-zero elements. HuBERT features were used as input for SAE.

Figure 4: Relative improvement of sparse speech representations on the dev set for each task. h is the dimensionality for the latents, while k denotes the number of their non-zero elements. wav2vec 2.0 features were used as input for SAE.

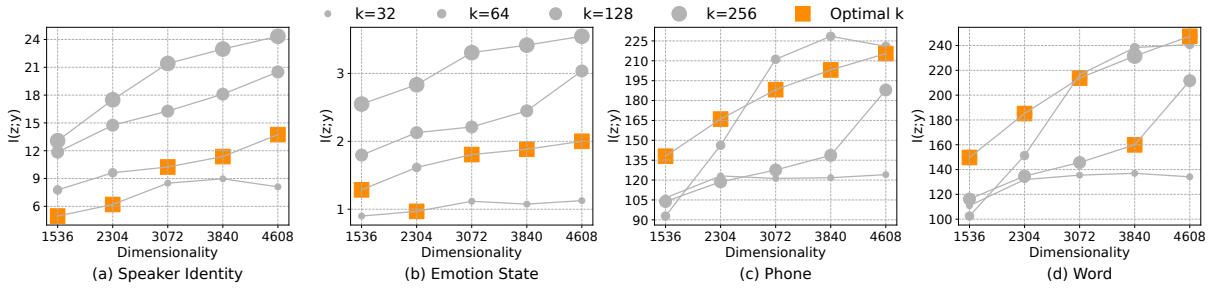

Figure 5: Mutual information in bits between latents and task-specific labels for speaker identity, emotion state, phone and word, respectively. HuBERT features were used as input for SAE.

found that $k = 32$ achieves the optimal performance for both layers as input, even if layer 0 has richer information about the speaker, but also noticed that the relative improvement for $k = 64$ is on par with the optimal one (51.22% vs 54.41%) for exp 3. By comparing exp 2 and 1, it is found that the optimal k has shifted from 256 to 128 after switching to layer 0, which suggests the impact of input features on the selection of optimal k even though both 256 and 128 indicate less sparsity. By comparing exp 5 and 6, it

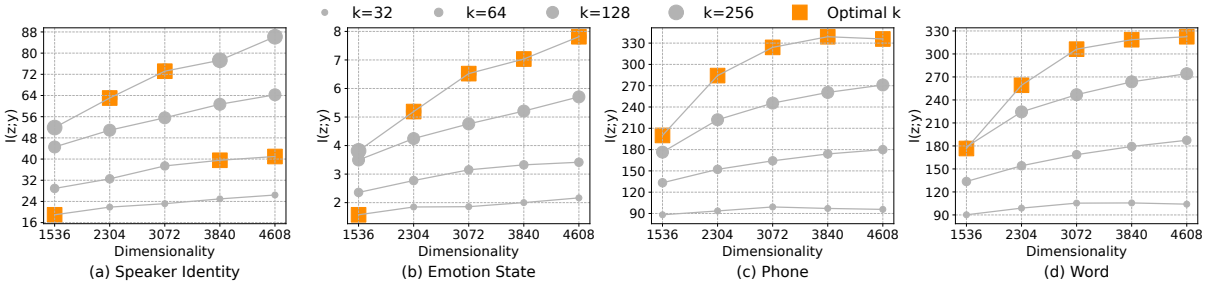

Figure 6: Mutual information in bits between latents and task-specific labels for speaker identity, emotion state, phone and word, respectively. wav2vec 2.0 features were used as input for SAE.

Exp ID	SSL features	Layer index	Task	$k = 32$	$k = 64$	$k = 128$	$k = 256$
1	WavLM	0	PR	-4.16	0.49	**1.96**	-0.84
2	WavLM	12	PR	-25.24	0.0	8.09	**10.02**
3	WavLM	0	SID	**54.41**	51.22	29.18	14.35
4	WavLM	12	SID	**12.72**	7.44	4.44	3.39
5	wav2vec 2.0	0	PR	-2.93	0.94	**3.00**	-2.03
6	wav2vec 2.0	12	PR	-61.10	-32.60	-22.80	-16.61

Table 6: Evaluation results of SID and PR with SAEs trained on different layers of wav2vec 2.0 and WavLM, respectively.

is found that switching to layer 0 can lead to positive performance improvement, even though layer 0 is noisier and richer in task-irrelevant information.