Self-supervised Rewiring of Pre-trained Speech Encoders: Towards Faster Fine-tuning with Less Labels in Speech Processing

Hao Yang[*] **Jinming Zhao**[*] **Gholamreza Haffari** **Ehsan Shareghi**
Department of Data Science & AI, Monash University
firstname.lastname@monash.edu

Abstract

Pre-trained speech encoders have facilitated great success across various speech processing tasks. However, fine-tuning these encoders for downstream tasks require sufficiently large training data to converge or to achieve state-of-the-art. In text domain this has been partly attributed to sub-optimality of the representation space in pre-trained Transformers. In this work, we take a sober look into pre-trained speech encoders and rewire their representation space without requiring any task-specific labels. Our method utilises neutrally synthesised version of audio inputs along with frame masking to construct positive pairs for contrastive self-supervised learning. When it is used for augmenting the WAV2VEC 2 encoder, we observe consistent improvement of isotropy in the representation space. Our experiments on 6 speech processing tasks, exhibit a significant convergence speedup during task fine-tuning as well as consistent task improvement, specially in low-resource settings.[1]

1 Introduction

Self-supervised pre-trained speech encoders (Hsu et al., 2021a; Baevski et al., 2020) are universal models that are beneficial to a wide range of speech processing tasks and domains (Liu et al., 2022; Tsai et al., 2022). Similar to other modalities such as text, these pre-trained encoders are fine-tuned towards downstream tasks (Wang et al., 2022; Gállego et al., 2021). While the fine-tuning step often benefits substantially from the presence of warm pre-trained data encoders, for involved tasks such as Automatic Speech Recognition (ASR), it still requires both sufficiently large training sets and several iterations (Yang et al., 2021) for convergence to an acceptable task performance.

Side-stepping the size of the parameter space as a well-studied challenge for fine-tuning Transformer models, a confounding factor contributing to this issue, which has been recently discussed for text domain (Su et al., 2022; Gao et al., 2021b; Liu et al., 2021; Su et al., 2021), is the sub-optimal utilisation of the representation space (e.g., anisotropy (Ethayarajh, 2019)). This is of paramount importance since speech, unlike text, carries information (e.g., prosodic and para-linguistic) beyond content which demands a richer utilisation of the representation space (Mohamed et al., 2022). Inevitably, less expressive initial representations translate into longer training and call for more labelled data, even in cases of frozen models. Nonetheless, understanding representation space utilisation in pre-trained speech Transformers is heavily under-explored (Pasad et al., 2021; Hsu et al., 2021b).

We move towards addressing this gap by highlighting the properties of such representation spaces, and proposing a self-supervised learning method that improves their utilisation prior to task fine-tuning. Our contrastive learning framework constructs positive pairs by (i) encouraging invariance to local perturbations both at the input and representation levels, and (ii) enhancing sensitivity to content by using monotonically synthesised version of speech inputs.

Our experimental findings across 6 diverse speech processing tasks (covering content, speaker and semantics tasks), built on top of the widely used WAV2VEC 2 LARGE (W2V2) (Baevski et al., 2020) encoder, demonstrate that contrastive rewiring brings substantial improvement, both in task performance and fine-tuning speed. Particularly, our approach shines in the low-resource condition, outperforming the W2V2 baseline with substantially fewer number of fine-tuning updates. For instance, in ASR with 1% training data, our approach achieves $1/4$ of the error in $1/5$ of fine-tuning updates. Beyond task performance and con-

[*] These authors contributed equally to this work.
[1] Our code and models are available at https://github.com/YangHao97/rewireW2V2.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 1952–1959
December 7-11, 2022

vergence speed, both our qualitative and quantitative analyses on the representation space highlight the improvements injected by our rewiring strategy.

2 Self-Supervised Contrastive Rewiring

Our method builds on top of a pre-trained speech encoder, by using a small (less than $7k$) set of raw unlabelled audio signals to form the self-supervised learning basis for contrastive rewiring. In what follows, we detail how utterance-level speech representations are produced from the underlying encoder, and provide a brief overview of the InfoNCE objective function used for our contrastive rewiring. We finish by explaining how we construct the pairs needed for contrastive learning.

Speech Representation. Most pre-trained speech encoders, including w2v2, do not have an explicit token representing utterance-level representation (e.g., [CLS] for BERT (Kenton and Toutanova, 2019)). Given a raw audio sequence s of length L, w2v2 emits m vectors, where $m \ll L$, at each layer (total of 24 Transformer layers + 1 feature extractor layer). Similar to Chung et al. (2021), we take the mean of these vectors to construct the utterance-level representation used for contrastive learning.

InfoNCE. We use the InfoNCE objective (Oord et al., 2018) to rewire speech representations by pulling positive examples, (s_i, s'_i), closer and pushing away the negative pairs, (s_i, s_j). The loss for a batch b of size $|\mathcal{D}_b|$ is,

$$\mathcal{L} = -\sum_{i=1}^{|\mathcal{D}_b|} \log \frac{\exp(\texttt{cos}(f(s_i), f(s'_i))/\tau)}{\sum\limits_{s_j \in N_i \cup \{s'_i\}} \exp(\texttt{cos}(f(s_i), f(s_j))/\tau)},$$

where $f(.)$ indicates the encoder, τ denotes the temperature hyperparameter, $\texttt{cos}(.,.)$ denotes the cosine similarity between two representations, N_i includes all negative examples for s_i. All parameters of the encoder are updated during optimisation.

2.1 Contrastive Pair Construction

Positive Pairs. We form positive pairs both at the raw and representation levels. For a given audio signal s_i, we deploy the following 3 strategies to construct its corresponding positive pairs, (s_i, s'_i):

`Twin`. Inspired by Liu et al. (2021); Gao et al. (2021b), given a speech sequence of length L, we first duplicate it. Then we randomly select a starting point for a span, and mask $p \times L$ consecutive signals from the audio, replacing them with `[MASK]`. We use $p = 20\%$ in our experiments. This is applied always and only once to each s_i.

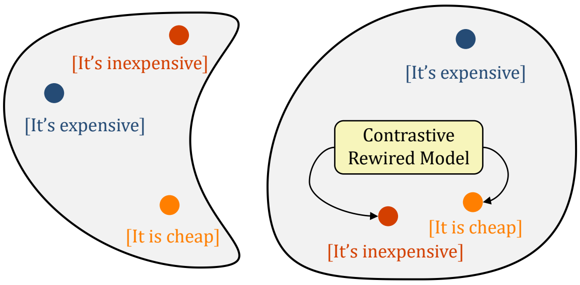

Figure 1: Conceptual visualisation: Vanilla representation space which is very sensitive to surface similarity of audio signals (left) vs. rewired representation space with `Neutral` strategy which places more emphasis on content similarity (right).

`Neutral`. For a given audio s_i, its monotonic neutral version is created from available transcripts[2] using Festival Speech Synthesis System.[3] The synthesizer is chosen because it is able to produce non-expressive speech, as demonstrated in previous studies (Lotfian and Busso, 2017). The neutral version is devoid of noise, prosody and para-linguistic features, focusing mostly on content. Figure 1 illustrates a visualisation of the desired expected effect from `Neutral` rewiring.

`Mixed`. While the `Twin` strategy aims to make the representations invariant to local changes and noise, the `Neutral` approach tends to rewire the space based on content-level similarity. To leverage the benefits of both worlds, as our main strategy, we uniformly interchange `Twin` and `Neutral` in the `Mixed` setting.

Negative Pairs. In all strategies, given a batch b and a sample $s_i \in b$, the set N_i of negative examples for s_i is $N_i = \{s_j | s_j \in b, j \neq i\}$. Further, we have specific negative samples added to N_i per each strategy to construct negative pairs, (s_i, s_j):

`Twin`. $N_i \cup \{\texttt{twin}(s_j) | s_j \in b, j \neq i\}$.

`Neutral`. $N_i \cup \{\texttt{neutral}(s_j) | s_j \in b, j \neq i\}$.

`Mixed`. Union of the above two.

Similar to Liu et al. (2021) and Gao et al. (2021b), in all our strategies, we apply dropout to perturb

[2] Alternatively, one can apply an off-the-shelf ASR first over speech to produce transcripts when transcripts are absent.

[3] http://festvox.org/festival

both the representations and internal components of Transformer. Note that Twin and masking are aligned with how a Transformer-based model is trained, so rewiring with this strategy is unlikely to create conflicts. Furthermore, Neutral potentially pushes representations towards eliminating paralingual feature from speech representations, as it shifts the focus to content.

3 Experiments

In this section we describe our experimental settings (§3.1) followed by downstream task results in full and low-resource scenarios (§3.2). We finish by providing an analysis on the quantitative and qualitative properties that were improved via our contrastive rewiring approach (§3.3).

3.1 Experimental setups

Rewiring Dataset. For all contrastive rewiring strategies on top of a pretrained w2v2, we used a small subset (6.6k instances) of LibriSpeech (Panayotov et al., 2015) train-clean-100 where instances lengths are under 180k, which is also part of training data for W2V2. LibriSpeech also contains transcripts, which we used for neutral rewiring to produce neutral speeches.

Downstream tasks. We experimented with 6 diverse downstream speech processing tasks from SUPERB benchmark (Yang et al., 2021): Automatic Speech Recognition (ASR), Speaker Diarization (SD), Intent Classification (IC), Slot Filling (SF), Keyword Spotting (KS) and Query by Example Spoken Term Detection (QbE). These tasks cover semantic, speaker, and content in speech tasks. For each task, we simulate various resource conditions by sampling 1%, 5%, 10% of the entire training set, while using the original dev set to decide the best model. Original test sets were used for evaluation. Each task is evaluated using its specific evaluation metrics. For details on tasks and evaluation metrics, see *Appendix* A. We detail data statistics in Table 1. We follow the instructions in SUPERB and use the s3prl toolkit[4] to prepare the datasets.

Baseline. While our approach is not dependent on a specific speech Transformer, we use w2v2 as the most widely used Transformer-based speech encoders. We follow the SUPERB evaluation pipeline by freezing it as an encoder for downstream tasks, while attaching a benchmark-specified lightweight prediction head for each task. For details on task head architectures, see *Appendix* A. We follow identical protocol for fine-tuning and evaluating all models.

Implementation details. During rewiring, we set the dropout to 0.1, the learning rate to 1e-6 and the temperature for InfoNCE to 0.04. To overcome the hardware constraint, we truncated audio signals and encoded them sequentially. To avoid memory issues that come from lengthy audio signals, we read one audio data at a time and set the audio length threshold to 90k. Our batch size for rewiring was 8, and we rewired w2v2 for 1.7k, 11.6k, 5k updates for Twin, Neutral and Mixed, respectively.

During downstream task fine-tuning, under 100% condition, we set the max step to $\sim 200k$ for both the baseline and our models. Under the 1%, 5% and 10% resource settings, we fine-tune the models for substantially less number of updates, although the baseline model still requires several steps for convergence (See Table 2). We follow the settings in Yang et al. (2021) for hyper-parameters, and use weighted-sum of hidden states of each layer from w2v2 as the final representation for downstream tasks. For more details, please refer to *Appendix* B.

3.2 Main Results

Our task fine-tuning results are presented in Table 2. The results highlight the improvement our contrastive rewiring brings across various downstream tasks, both in performance and fine-tuning speed. Notably, in 1% and 5% training scenarios, our models (at least 1 and in many cases all 3 strategies) outperform the w2v2 baseline on all tasks, while requiring substantially less number of fine-tuning updates. More details per task follows:

IC The benefit of rewiring in the low-resource (1%) condition is massive. Even at 100%, our method outperforms W2V2 without rewiring. The best rewiring setting helps to speedup convergence by an average of 22× in various data conditions.

SF Similarly, rewiring is beneficial in all settings, even at 100%. As the training data size decreases, the benefit of rewiring increases.

SD Our method is helpful in low- and mid-resource settings; at 100% it has lower performance, but Mixed helps the model to converge 6× faster.

KS Rewiring boosts performance and convergence

[4] https://github.com/s3prl/s3prl

				# Training Instances			
Task	Eval.	Type	Avg.Src Len	1%	5%	10%	100%
SD	DER↓	speaker	233k	0.14k	0.70k	1.39k	13.9k
SF	F1↑	semantic	45.6k	1.05k	5.24k	10.5k	104.7k
IC	Acc↑	semantic	36.6k	0.23k	1.16k	2.32k	23.1k
KS	Acc↑	content	15.7k	0.51k	2.56k	5.11k	51.1k
ASR	WER↓	content	201k	0.28k	1.43k	2.86k	28.5k
QbE	MTWV↑	content	107k	∗	∗	∗	∗

Table 1: Dataset statistics. ∗: QbE is zero-shot. Tasks types are determined by SUPERB.

		SD		SF		IC		KS		ASR	
Tr.	Model	DER ↓	🕐	F1 ↑	🕐	Acc ↑	🕐	Acc ↑	🕐	WER ↓	🕐
1%	W2V2	10.23	6.7k	57.88	42k	12.54	6.8k	85.17	9.2k	99.99	50k
	Twin	9.04	**0.1k**	65.14	38k	7.03	**0.2k**	85.98	2k	23.43	12k
	Neutral	12.73	0.4k	63.05	36k	35.48	0.4k	93.08	**0.25k**	36.20	12k
	Mixed	12.30	0.2k	64.04	**17k**	43.05	0.8k	90.58	1.3k	26.35	**10k**
5%	W2V2	9.20	4k	78.29	58k	53.07	16k	94.25	20k	14.70	100k
	Twin	8.15	0.4k	82.65	**56k**	39.41	**2.8k**	94.12	3.75k	8.77	**46k**
	Neutral	10.01	1.2k	79.48	62k	91.27	4.3k	94.28	1.75k	21.12	40k
	Mixed	10.39	**0.2k**	81.00	68k	90.29	3.1k	94.77	**1.25k**	14.25	48k
10%	W2V2	8.21	6k	80.74	90k	77.91	45k	95.85	15.5k	5.96	**90k**
	Twin	7.70	1.6k	85.00	88k	57.97	**1.6k**	94.97	4.5k	6.45	92k
	Neutral	8.57	2.4k	82.30	84k	92.75	15k	94.90	4.5k	16.72	98k
	Mixed	9.60	**0.6k**	84.02	**74k**	93.67	2k	95.07	**1.5k**	11.27	93k
100%	W2V2$^\diamond$	5.68	27.6k	87.67	150k	94.60	90k	96.64	55k	3.78*	166k
	Twin	5.95	6.5k	89.93	160k	93.36	55k	96.62	**15k**	4.07*	**80k**
	Neutral	6.67	11k	88.05	**125k**	97.18	15k	96.30	20k	10.34*	105k
	Mixed	6.73	**4.5k**	89.18	140k	97.18	**5k**	96.75	25k	7.46*	110k
QbE (100%) : [w2v2$^\diamond$:4.93], [Twin:5.04], [Neutral:3e-10], [Mixed: **7.50**]											

Table 2: Results and the number of fine-tuning updates to achieve the best performance for downstream tasks in various resource conditions. *: for ASR the number of training instances was 76%, as remaining overlapped with the data used for rewiring. $^\diamond$: replicated.

significantly at 1%. It has the similar performance in other conditions, but still helps training with an average speedup of 10×.

ASR Our Mixed model at 1% achieves an error rate of 26 after 10k updates, whereas w2v2 has an error rate of 99.99 even after 50k updates. The gain from rewiring is also noticeable at 5%, while levelling out afterwards.

QbE Rewiring improves performance significantly with our Mixed model yielding 7.50 in MTWV over 4.93 produced by the vanilla W2V2.

Overall, our proposed approach improves performance and convergence across various speech processing tasks in all resource conditions, with more remarkable gains in low- to mid-resource conditions.

3.3 Analysis and Discussion

Qualitative Analysis. Figure 2 demonstrates the t-SNE (van der Maaten and Hinton, 2008) visualisation of the impact of applying each strategy to w2v2. We clearly observe that better clustering of the representation space, specially after applying neutral (bottom-lsumeft), emerges without any task fine-tuning.

Task Type. According to Table 1, we have 3 types of tasks, content, semantic and speaker. As expected the content and semantic tasks are the biggest gainers in the low-resource setting (1-5%) from our contrastive rewiring. This verifies our earlier qualitative analysis and aligns well with the motivation for leveraging neutral speeches in the Neutral strategy which is expected to put more emphasise on content.

Isotropy. We speculate that the benefits of our rewiring strategies also roots in reshaping the representation space geometry. The isotropy of the embedding space is a desired property and we conjecture that the representations of w2v2 are potentially anisotropic: crowded into a narrow slices of the representation space. To test our conjecture,

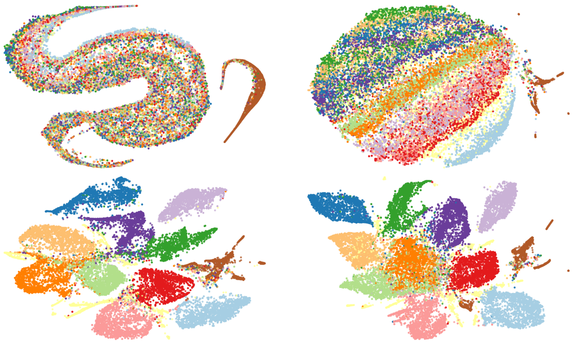

Figure 2: t-SNE visualisations of the representations from Keyword Spotting (Yang et al., 2021) training set prior to fine-tuning. **Top-Left:** W2V2; **Top-Right:** `Twin`; **Bottom-Left:** `Neutral`; **Bottom-Right:** `Mixed`. Colours indicate class labels.

we calculated the isotropy scores of speech representations produced by 4 models on 5 different datasets. We approximate the isotropy score (Mu and Viswanath, 2018),

$$IS(\mathcal{V}) = \frac{\min_{m \in \mathcal{M}} \sum_{v \in \mathcal{V}} \exp(m^\intercal v)}{\max_{m \in \mathcal{M}} \sum_{v \in \mathcal{V}} \exp(m^\intercal v)},$$

where $\mathcal{V}$ is the matrix of representations, and $\mathcal{M}$ is the set of eigen vectors of $\mathcal{V}^\intercal \mathcal{V}$. The isotropy scores of W2V2, `Twin`, `Neutral`, and `Mixed` models are on the order of 1e-300, 1e-10, 1e-30, and 1e-10 respectively. This result confirms our conjecture that our three models improve isotropy of the representation space by orders of magnitude compared with W2V2.

4 Conclusion and Future Work

In this paper, we presented effective and efficient self-supervised contrastive learning methods to rewire the representations of speech pre-trained Transformer model. We demonstrated that lightly rewiring WAV2VEC 2 improves the convergence speed of fine-tuning as well as task performance on 6 downstream tasks. In particular, in low-resource condition our method performed substantially better than the underlying WAV2VEC 2. Our analysis indicated the rewiring has created a much better discriminated representation space, making it better suited for fine-tuning towards tasks. As future work, we plan to cover more downstream tasks, and invest more into designing hard negative pairs to further augment the contrastive learning.

5 Limitations

We hoped to extend our experiments to all tasks on SUPERB but certain tasks involved data access difficulties (we initiated the requests but never got access). Additionally, we did not see significant gain with rewiring on Phoneme Recognition, which could stem from our construction of utterance-level representation. This suggests finer grained granularity of representations need to be included for fine-grained tasks. It also requires further investigation to fully understand why our method performs extremely well in certain source conditions and relatively well on certain tasks compared to others. Although we have provided certain conjectures, this analysis requires a standalone work.

6 Ethics Statement

Our work is built on top of WAV2VEC 2, which is pretrained on massive speech data. Our goal was not to attend to alleviate the well-documented issues (e.g., privacy, undesired biases, etc) that large pretrained models have. For this reason, we share the similar potential risks and concerns posed by these models.

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.

Yu-An Chung, Yonatan Belinkov, and James Glass. 2021. Similarity analysis of self-supervised speech representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3040–3044. IEEE.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Gerard I Gállego, Ioannis Tsiamas, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: Upc at iwslt 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021a. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 316–321.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

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

Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, et al. 2021b. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459.

Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabeleiro, Kun Qian, Xin Jing, Alexander Kathan, Bin Hu, and Bjoern W Schuller. 2022. Audio self-supervised learning: A survey. *arXiv preprint arXiv:2203.01205*.

Reza Lotfian and Carlos Busso. 2017. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *arXiv preprint arXiv:2205.10643*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

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

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.

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

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. 2022. TaCL: Improving BERT pre-training with token-aware contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2497–2507, Seattle, United States. Association for Computational Linguistics.

Hsiang-Sheng Tsai, Heng-Jui Chang, Wen-Chin Huang, Zili Huang, Kushal Lakhotia, Shu-wen Yang, Shuyan Dong, Andy Liu, Cheng-I Lai, Jiatong Shi, et al. 2022. Superb-sg: Enhanced speech processing universal performance benchmark for semantic and generative capabilities. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8479–8492.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. 2022. Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7097–7101. IEEE.

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, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 1194–1198. ISCA.

A SUPERB Tasks Details

We provide a brief overview of the tasks, prediction head settings and evaluation metrics. For further details please refer to SUPERB paper (Yang et al., 2021) or the SUPERB leaderboard.[5]

- ASR aims to transcribe audio into text. A vanilla 2-layer BLSTM is applied as the downstream task model, optimised with CTC loss. The evaluation metric is word error rate (WER).
- KS classifies utterances to detect specific keywords. Mean-pooling and a linear layer with cross-entropy loss are applied as the downstream task model. The task is evaluated using accuracy (ACC).
- QbE aims to detect spoken terms in an audio database by calculating whether a given query matches a spoken document. It does not require training. Dynamic Time Warping(DTW) and standard distance functions are used on all hidden states to report the final score. Maximum term weighted value (MTWV) is used for evaluation.
- SD, given an audio in which more than one person speak alternately, aims to determine the speaker at each timestamp. A single-layer 512-unit LSTM is applied as the downstream task model to SD task. The evaluation metric is diarisation error rate (DER).
- IC is designed to detect the intent of speaker from a spoken utterance. Mean-pooling and a linear transformation with cross-entropy loss are employed in the downstream task model. The evaluation metric is accuracy (ACC).
- SF aims to detect a sequence of semantic slot-types based on spoken words. Slot-type labels are included in transcriptions as special tokens, while SF is treated as an ASR problem. A vanilla 2-layer BLSTM with CTC loss is applied as the downstream task model. The evaluation metrics are slot-type F1 score and slot-value CER.

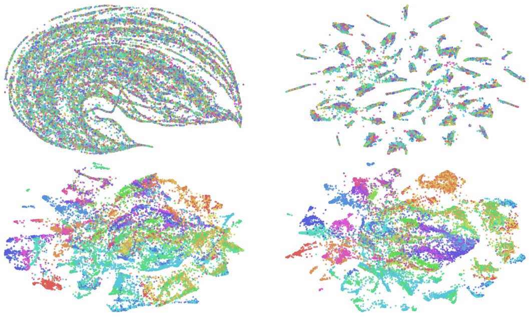

Figure 3: t-SNE visualisations of the representations from Intent Classification (Yang et al., 2021) training set prior to fine-tuning. **Top-Left:** W2V2; **Top-Right:** `Twin`; **Bottom-Left:** `Neutral`; **Bottom-Right:** `Mixed`. Colours indicate class labels.

	Training Instances		
Task	1%	5%	10%
SD	20k	20k	50k
SF	50k	100k	100k
IC	20k	20k	50k
KS	20k	50k	50k
ASR	50k	100k	100k

Table 3: Maximum number of training steps.

B Implementation Details

We rewire WAV2VEC 2 for 1.7k, 11.6k, 5.0k updates (≈1, 7, 3 epochs) for TWIN, NEUTRAL and MIXED, respectively. Applying more epochs may lead to overfitting, for TWIN, training loss drops to almost zero in epoch 2. Next, in downstream tasks, under the full-resource condition, we set the max step to ∼200k for both the baseline and our models. Under the 1%, 5% and 10% resource settings, Table 3 shows the max step we set for training models respectively, and test the best checkpoints accordingly. Note that the baseline and our models are trained with the same number of updates in all settings. During the rewiring process, we use dropout = 0.1 of WAV2VEC 2 for all training instances. The learning rate is set to 1e-6 and the temperature for the infoNCE loss to 0.04. Additionally, it is perceived that contrastive learning requires a sufficient number of positive and negative pairs (Gao et al., 2021a), which cannot be achieved naively, again, due to the length issue. To solve the problem, we use a batch size of 4. For each input within the batch, we get the augmented version and feed the two data points to the network to obtain two utterance-level vectors; this process applies to the rest of 3 training examples. This effectively relaxes the memory requirement

[5] https://superbbenchmark.org/leaderboard

for training pretrained speech models, given the hardware constraint. To avoid memory issues that come from lengthy audio signals, in practice we read one audio data at a time and set the audio length threshold to 90k. Whenever the audio length is greater than the threshold, we split the audio input into two parts with equal lengths, and randomly select one of them as the basis for further augmentation. Furthermore, to mask the augmented data in training TWIN, we randomly pick a starting point in the first four-fifths of the frames of the audio data, and then mask consecutive frames that are one-fifth of its total length. For NEUTRAL, we generated neutral speech from the transcriptions of the chosen LibriSpeech subset with the aforementioned TTS software offline. While training MIXED, we randomly select a augmentation technique from *twin* and *neutral* to construct positive pairs. The truncation trick applies to all three methods.

C t-SNE on Intent Classification

Figure 3 illustrates t-SNE (van der Maaten and Hinton, 2008) visualisation on IC. While the representation space is not as well-separated as KS, it is still rather clear that NEUTRAL and MIXED have better separation of speech representations and consistency within clusters (higher concentration of same color within clusters). While the points seem to be separated across the space, a closer look indicates that the shaped clusters are substantially mixed, making it much more difficult for the task layer to discriminate between these points (much worse than W2V2).