Speech Representation Learning through Self-supervised Pretraining and Multi-task Finetuning

Yi-Chen Chen[1], Shu-wen Yang[1], Cheng-Kuang Lee[2], Simon See[2], Hung-yi Lee[1]

[1]National Taiwan University, Taiwan
[2]NVIDIA AI Technology Center, NVIDIA
[1] {f06942069, r08944041, hungyilee}@ntu.edu.tw, [2]{ckl,ssee}@nvidia.com

Abstract

Speech representation learning plays a vital role in speech processing. Among them, self-supervised learning (SSL) has become an important research direction. It has been shown that an SSL pretraining model can achieve excellent performance in various downstream tasks of speech processing. On the other hand, supervised multi-task learning (MTL) is another representation learning paradigm, which has been proven effective in computer vision (CV) and natural language processing (NLP). However, there is no systematic research on the general representation learning model trained by supervised MTL in speech processing. In this paper, we show that MTL finetuning can further improve SSL pretraining. We analyze the generalizability of supervised MTL finetuning to examine if the speech representation learned by MTL finetuning can generalize to unseen new tasks.

1 Introduction

Recently many SSL approaches have been proposed for pretraining models for speech processing tasks, including pretraining with generative losses (Chung et al. 2019; Liu et al. 2020; Liu, Li, and Lee 2021; Ling and Liu 2020) or discriminative losses (Oord, Li, and Vinyals 2018; Schneider et al. 2019; Baevski, Schneider, and Auli 2020; Baevski et al. 2020; Hsu et al. 2021). Some approaches use multi-task learning with multiple SSL objectives (Pascual et al. 2019; Ravanelli et al. 2020). After a shared model is pretrained with SSL to extract general representations, it can then be specialized on downstream tasks with task-specific head models and simple finetuning. This method achieves state-of-the-art performance in many applications.

To fairly evaluate the generalizability of SSL approaches without further heavy downstream task-specific finetuning, the SUPERB benchmark (Yang et al. 2021) is proposed. The SUPERB benchmark measures the performance of a shared model across a wide range of speech processing tasks without heavy finetuning. Ten tasks are included to investigate four aspects of speech: content, speaker, semantics, and paralinguistics. To evaluate a general model trained with SSL, the pretrained model parameters are frozen, and the fixed representations are extracted and fed into each task-specific prediction head (small downstream model) for training. During the evaluation, the pretrained shared model and trained prediction heads are used on all tasks. In the above sce-

nario, some self-supervised models show outstanding performances on all the ten tasks in SUPERB.

Supervised multi-task learning (MTL) is to train a shared model on various downstream tasks (Kendall, Gal, and Cipolla 2018; Chen et al. 2018; Sener and Koltun 2018; Yu et al. 2020; Wang et al. 2020). This paper wants to investigate if MTL on various downstream tasks can further improve the representations from SSL. In computer vision (CV) (Silberman et al. 2012; Everingham et al. 2010) and natural language processing (NLP) (McCann et al. 2018; Wang et al. 2018), general models trained by MTL approaches can be evaluated on benchmarks that include various tasks. However, in speech, there has not been a systematic study of general representation learning models trained by MTL of various speech processing tasks. SpeechNet (Chen et al. 2021) proposes a general modularized model to perform a variety of speech processing tasks, but the purpose of SpeechNet is to allow different tasks to utilize different subsets of the general modules in SpeechNet, rather than proposing one shared model to extract general representations for training task-specific heads of downstream tasks.

In this paper, we investigate two MTL training scenarios and also one task transfer learning scenario. For the two MTL scenarios, we select a state-of-the-art SSL pretrained shared model in the SUPERB benchmark as the starting point for MTL. Then we train the shared model with MTL in two different scenarios:

- All-task MTL Finetuning: Finetune the SSL pretrained shared model with all tasks in SUPERB. It serves as a strong baseline for SSL approaches and the following scenarios.

- Leave-one-out MTL Finetuning: Finetune the SSL pretrained shared model with all but one tasks in SUPERB. We can observe the influence of removing one task on the learned representations and their performance on the other tasks.

To further examine if the representations learned with supervised MTL can generalize to an unseen new task, we have an additional Task Transfer Learning scenario.

- We take a shared model from the Leave-one-out MTL scenario and freeze its parameters. We extract the representations with this shared model for training the prediction head of the remaining task that is not involved in

MTL finetuning.

Through the different training scenarios above, we perform a preliminary study of the generalizability of representation learning by MTL of various speech processing tasks on a standard benchmark. The code is released for reproduction and future extension. [1].

Since the labels from the downstream datasets are used to finetune the pretrained and shared SSL model using supervised MTL, this work does not intend to follow the SUPERB rules strictly.

2 Training Scenarios

In this section, we describe four related training scenarios in the following subsections: SSL Pretraining, All-task MTL Finetuning, Leave-one-out MTL Finetuning, and Task Transfer Learning. The two MTL finetuning scenarios require the SSL pretrained shared model, and the Task Transfer Learning scenario requires the shared model from the Leave-one-out MTL Finetuning scenario.

2.1 SSL Pretraining

Many SSL approaches are evaluated and compared on the ten tasks in SUPERB (Yang et al. 2021). For each SSL approach, we reuse an existing pretrained model that is a baseline already reported on in the SUPERB paper. Then we use this pretrained model as the shared model to extract representations for all downstream tasks. The parameters of the pretrained model are frozen. Then we train each task-specific prediction head (small downstream model) with the fixed representations, as illustrated in Figure 1 (a).

2.2 All-task MTL Finetuning

We take the shared model pretrained in Subsection 2.1 as the starting point. Then we further finetune the shared model with MTL by jointly training it with downstream task-specific heads of all tasks in SUPERB, as illustrated in Figure 1 (b-1). In this way, the shared model can be updated by the gradients of all tasks to fit the respective objectives of each task. Therefore, the representations extracted from the shared model can perform well on the tasks involved in MTL. It serves as a strong baseline for SSL approaches and the following scenarios. To further examine the generalizability of representation learning by supervised MTL, we have two additional training scenarios below.

2.3 Leave-one-out MTL Finetuning

Similarly, we take the shared model pretrained in Subsection 2.1 as the starting point. Then we further finetune the shared model with MTL by jointly training it with downstream task-specific heads of **all but one** tasks in SUPERB, as illustrated in Figure 1 (b-2). Compared to the finetuned shared model in Subsection 2.2, we can observe the influence of removing one task on the learned representations and their performance on the other tasks.

2.4 Task Transfer Learning

We take the finetuned shared model in Subsection 2.3 as the pretrained shared model in this scenario. Then we freeze the shared model, and only train the downstream head model of the remaining task that is not used in MTL finetuning, as illustrated in Figure 1 (c).

Comparing the representations in this scenario with those learned with only SSL approaches in Subsection 2.1, we can observe the generalizability of the representations learned with MTL on a new task compared to SSL only. On the other hand, in comparison with the representations learned with All-task MTL in Subsection 2.2, we can observe how the performance of a task is influenced if this task is not involved in the MTL finetuning.

3 Experimental setup

3.1 Tasks In SUPERB

Ten tasks in SUPERB can be used to investigate four aspects of speech: **content** (Phoneme Recognition (PR), Automatic Speech Recognition (ASR), Keyword Spotting (KS), and Query by Example Spoken Term Detection (QbE)), **speaker** (Speaker Identification (SID), Automatic Speaker Verification (ASV), and Speaker Diarization (SD)), **semantics** (Intent Classification (IC) and Slot Filling (SF)), and **paralinguistics** (Emotion Recognition (ER)). Since no downstream model training is required in QbE, we only perform MTL experiments and compare the results on the other nine tasks. For more details about the description of each task, please look at the original SUPERB paper. As for the datasets and splits used for each task, we follow the original settings in SUPERB (PR (Panayotov et al. 2015), ASR (Panayotov et al. 2015), KS (**?**), SID (Nagrani et al. 2020), ASV (Nagrani et al. 2020), SD (Cosentino et al. 2020), IC (Lugosch et al. 2019), SF (Lai et al. 2021), and ER (Busso et al. 2008)).

3.2 The SSL Pretraining Approach In Experiments

Many SSL approaches are evaluated and compared on the tasks in SUPERB. Among them, HuBERT (Hsu et al. 2021) achieves the overall best performance. Therefore, we select HuBERT as the SSL pretraining approach across all of our experiments.

HuBERT utilizes an offline clustering algorithm on hidden representations to provide aligned target labels for a BERT-like (Devlin et al. 2019) prediction. The clustered labels at the masked locations serve as the prediction targets. We use a weighted sum of hidden representations of all layers in the HuBERT model as the representations for downstream heads, as in SUPERB.

3.3 Model Architecture and Implementation Details

Since MTL requires more computational resources than single-task training, we adopt HuBERT Base rather than Large in SUPERB as our shared model architecture. For task-specific head architectures, we simply follow the settings in SUPERB. We use a batch size of 2 for MTL finetuning in training scenarios described in Subsections 2.2 and

---

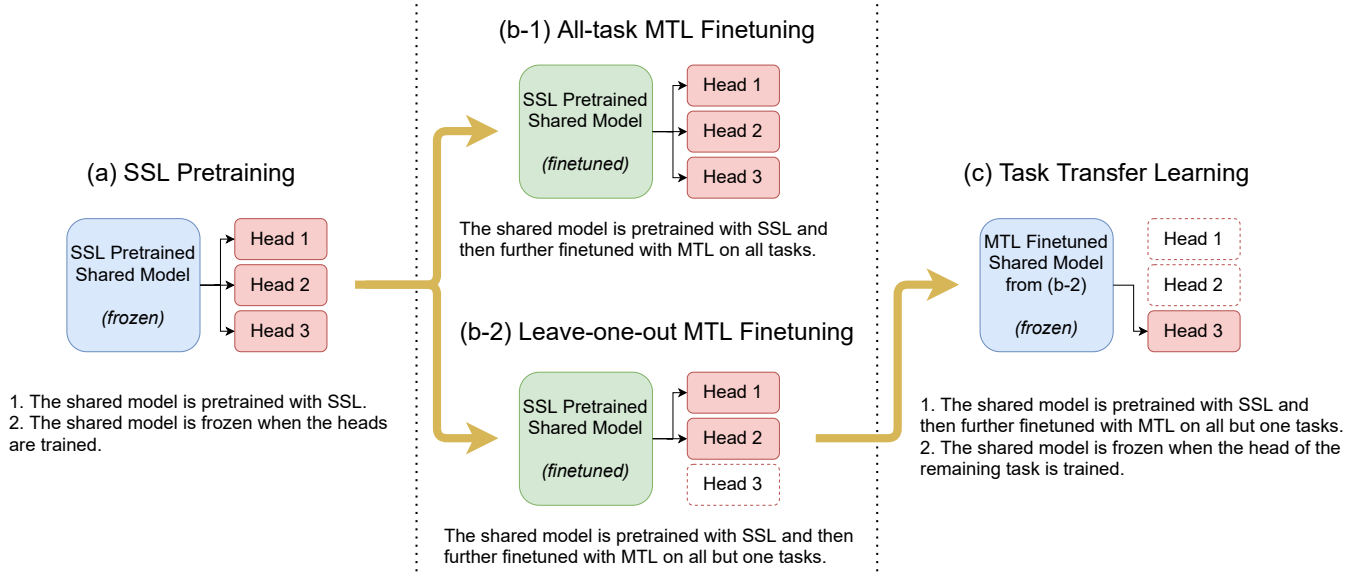[1]https://github.com/s3prl/s3prl/tree/multi-task-distributed

Figure 1: Four different training scenarios. Scenarios (b-1) and (b-2) require the pretrained shared model from (a). Scenario (c) requires the finetuned shared model from (b-2). The parameters of the shared model in scenarios (a) and (c) are frozen.

2.3, and a batch size of 8 for downstream head training in the training scenario described in Subsection 2.4. Each model is trained with an Adam optimizer with a linearly warmup learning rate from 0 to 1e-5 for the first 5000 steps and then a linearly decaying learning rate to 0 for 195000 steps. For more implementation details, please refer to the released code.

4 Experimental Results

All experimental results are presented in Table 1. An up-/down-arrow beside an evaluation metric means that better performance results in a higher/lower number of that metric. The results are grouped according to four different training scenarios corresponding to the subsections in Section 2 respectively.

4.1 The Performance Of All-task MTL Finetuning

From the comparison between Scenario (a) and Scenario (b-1), we observe that the performance of the shared model finetuned with All-task MTL is better in the tasks ASR, PR, SD, ER, IC, KS, SID, and worse only in the tasks SF and ASV than SSL pretraining. It indicates that MTL is a strong baseline for SSL pretraining or other representation learning approaches.

One thing worth noting is that all tasks except for ASV do not suffer from overfitting in terms of the validation scores during training. However, the model is prone to overfit on ASV. In Scenario (a), the model checkpoint of a downstream head can be determined by the best validation score during training for each task respectively. Yet in Scenario (b-1), since the shared model is jointly trained with the downstream heads of all tasks, it is hard to select the model checkpoint based on the validation scores of all tasks. In this paper,

we simply select the last model checkpoint after 200,000 training steps for all tasks. It may be a reason for the worse performance of MTL finetuning on ASV. We leave exploring a better method to select the model checkpoint with MTL in future work.

4.2 The Influence Of Removing One Task In MTL Finetuning

The rows in Scenario (b-2) show the results of finetuning the shared model with Leave-one-out MTL. To compare these results with Scenario (b-1) more clearly, we have the numbers in Scenario (b-2) be bold or underlined if they are better or worse than (b-1) respectively. Furthermore, we calculate the relative score increases/decreases of Scenario (b-2) compared to Scenario (b-1), and plot two relation graphs in Figure 2 accordingly. For SF, we use CER to plot the edges because the relative changes are small in terms of F1. If a task A improves/regresses after removing another task B in MTL, it means that B can hurt/help A in MTL.

- For **ASR**, PR is an important auxiliary task for ASR, and SD helps a little. ASR performs better after removing any of the other tasks.

- For **PR**, ASR helps in MTL while the other tasks hurt the performance of PR. We can observe that content recognition tasks such as ASR and PR are hurt the most by speaker recognition tasks such as ASV and SID.

- For **SF**, ASR, PR, ER, and SID help in MTL. IC and KS help a little in terms of F1 but hurt in terms of CER. SD and ASV hurt SF.

- For **SD**, all of the other tasks hurt SD. SD is also hurt the most by speaker recognition tasks such as ASV and SID. Although SD, ASV and SID are all related to speaker

Table 1: Experimental Results of training scenarios described in Section 2. We have the numbers in Scenario (b-2) be bold or underlined if they are better or worse than (b-1) respectively.

Scenario	Tasks for MTL Finetuning	ASR WER↓	PR PER↓	SF F1↑	SF CER↓	SD DER↓	ER ACC↑	IC ACC↑	KS ACC↑	ASV EER↓	SID ACC↑
(a) SSL	N/A	6.42	5.41	88.53	25.20	5.88	64.24	98.34	96.30	5.11	81.42
(b-1) SSL+MTL	all	6.22	3.61	87.56	26.76	4.93	67.28	99.60	97.34	6.76	90.86
(b-2): SSL+MTL	all but ASR	X	3.63	87.28	27.11	**4.89**	65.07	**99.63**	**97.57**	7.78	90.69
	all but PR	6.79	X	86.94	27.66	**4.81**	66.73	**99.66**	**97.44**	7.94	**91.16**
	all but SF	**6.10**	**3.39**	X	X	**4.73**	65.71	99.58	97.18	7.61	90.70
	all but SD	6.28	**3.54**	**87.94**	**26.31**	X	66.73	**99.63**	97.11	7.49	90.79
	all but ER	**6.17**	**3.40**	87.45	26.90	**4.77**	X	99.55	97.27	7.19	90.51
	all but IC	**6.13**	**3.34**	87.65	26.94	**4.78**	66.08	X	97.27	**6.74**	90.55
	all but KS	**6.17**	**3.55**	87.83	26.88	**4.91**	66.27	**99.71**	X	7.86	90.67
	all but ASV	**5.90**	**2.79**	**87.88**	**26.52**	**3.61**	64.88	99.58	**97.44**	X	85.06
	all but SID	**5.95**	**3.25**	87.33	27.39	**4.50**	**68.66**	99.55	97.27	9.00	X
(c) Task Transfer	N/A	6.27	5.79	88.14	26.24	5.80	64.24	97.42	96.33	7.55	62.05
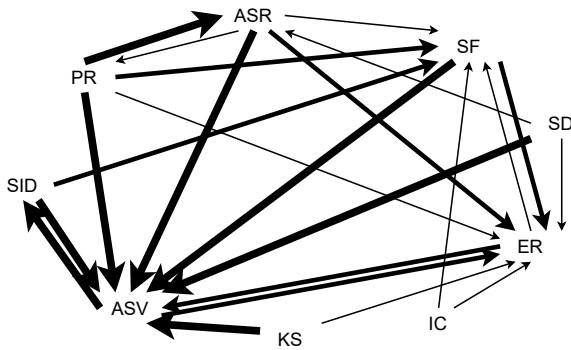
The improvement relations of tasks.
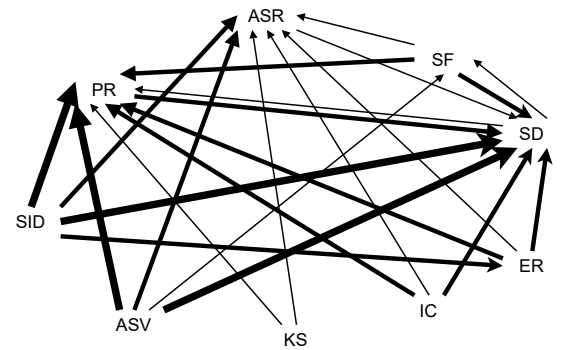
The hurt relations of tasks.

Figure 2: Two relation graphs of tasks. If a task A performs worse/better after removing a task B in MTL, we connect an edge from B to A in the improvement/hurt graphs, indicating B can improves/hurts the performance of A. The width of an edge is thin/medium/thick if the relative score change is in the range [0.5%, 2%), [2%, 8%), or larger than 8%. If the relative score change is less than 0.5%, we consider it negligible and no edge is connected.

characteristics, SID and ASV focus on the utterance-level embedding, while SD aims to distinguish frame-level speaker characteristics. Therefore, the fine-grained information needed by SD may be lost when jointly trained with ASV or SID.

- For **ER**, all of the other tasks except for SID help ER.
- For **IC** and **KS**, the influences of the other tasks are negligible.
- For **ASV**, all of the other tasks except for IC help ASV. As discussed in Subsection 4.1, ASV suffers from severe overfitting. Therefore, jointly learning ASV with other tasks can mitigate this issue, especially with SID.
- For **SID**, all of the other tasks except for PR help SID. ASV especially helps a lot while the others help a little.

From another perspective, the results may provide a different insight in addition to the evaluation of supervised MTL as a representation learning. If we focus on a certain primary task, we may select proper auxiliary tasks to assist the primary task based on these MTL experimental results. For example, if we want to have a better performance on ER, we can jointly train ER with all of the other tasks except for SID.

The transitive relations of tasks in MTL need to be further verified. For example, suppose we know jointly training a task A with another task B can improve the performance of A; and we also know jointly training A with another task C can improve the performance of A. Yet, we cannot conclude that jointly training A with B and C simultaneously can improve A's performance. We leave more in-depth research of MTL and its optimization on speech processing tasks in future work.

4.3 The Performance Of Task Transfer Learning

To further examine the generalizability of representation learning by MTL, we present the results of the Task Transfer Learning scenario in Table 1 (c). Compared to Table 1 (a), all of the tasks except for ASR and KS in Scenario (c) perform

worse. It indicates SSL pretraining is still a more generalizable representation learning approach for a new downstream task. On the other hand, compared to Table 1 (b-1), all of the tasks except for SF in Scenario (c) perform worse. It indicates whether a task is involved in MTL is crucial to the performance of this task.

To obtain better generalizability, it is worth trying to train the shared model with both SSL and MTL simultaneously as semi-supervised MTL representation learning. We leave this exploration in future work.

5 Conclusion And Discussion

In this work, we investigate different training scenarios of supervised MTL as a speech representation learning approach along with SSL pretraining on a benchmark with various speech processing tasks. We analyze the generalizability of representations learned with supervised MTL empirically.

This paper is only a preliminary study of MTL with various speech processing tasks. The performance of MTL is dependent on many factors, such as the amount of data, task relationships, noise and so on. These factors should be isolated and investigated with more theoretical analyses and empirical experiments in the future.

References

Baevski, A.; Schneider, S.; and Auli, M. 2020. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. In *ICLR*.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *NeurIPS*.

Busso, C.; et al. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4): 335–359.

Chen, Y.-C.; Chi, P.-H.; Yang, S.-w.; Chang, K.-W.; Lin, J.-h.; Huang, S.-F.; Liu, D.-R.; Liu, C.-L.; Lee, C.-K.; and Lee, H.-y. 2021. SpeechNet: A Universal Modularized Model for Speech Processing Tasks. *arXiv preprint arXiv:2105.03070*.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, 794–803. PMLR.

Chung, Y.-A.; Hsu, W.-N.; Tang, H.; and Glass, J. 2019. An Unsupervised Autoregressive Model for Speech Representation Learning. In *Interspeech*, 146–150.

Cosentino, J.; Pariente, M.; Cornell, S.; Deleforge, A.; and Vincent, E. 2020. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Hsu, W.-N.; Tsai, Y.-H. H.; Bolte, B.; Salakhutdinov, R.; and Mohamed, A. 2021. HuBERT: How much can a bad teacher benefit ASR pre-training? In *ICASSP*, 6533–6537. IEEE.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Lai, C.-I.; Chuang, Y.-S.; Lee, H.-Y.; Li, S.-W.; and Glass, J. 2021. Semi-Supervised Spoken Language Understanding via Self-Supervised Speech and Language Model Pretraining. In *ICASSP*.

Ling, S.; and Liu, Y. 2020. DeCoAR 2.0: Deep Contextualized Acoustic Representations with Vector Quantization. *arXiv preprint arXiv:2012.06659*.

Liu, A. T.; Li, S.-W.; and Lee, H.-y. 2021. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 2351–2366.

Liu, A. T.; Yang, S.-w.; Chi, P.-H.; Hsu, P.-c.; and Lee, H.-y. 2020. Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders.

Lugosch, L.; Ravanelli, M.; Ignoto, P.; Tomar, V. S.; and Bengio, Y. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Interspeech*, 814–818.

McCann, B.; Keskar, N. S.; Xiong, C.; and Socher, R. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Nagrani, A.; Chung, J. S.; Xie, W.; and Zisserman, A. 2020. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60: 101027.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*, 5206–5210.

Pascual, S.; Ravanelli, M.; Serrà, J.; Bonafonte, A.; and Bengio, Y. 2019. Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks. In *Interspeech*, 161–165.

Ravanelli, M.; Zhong, J.; Pascual, S.; Swietojanski, P.; Monteiro, J.; Trmal, J.; and Bengio, Y. 2020. Multi-task self-supervised learning for robust speech recognition. In *ICASSP*, 6989–6993.

Schneider, S.; Baevski, A.; Collobert, R.; and Auli, M. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Interspeech*.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. In *NeurIPS*, 525–536.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, 746–760. Springer.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *ICLR*.

Wang, Z.; Tsvetkov, Y.; Firat, O.; and Cao, Y. 2020. Gradient Vaccine: Investigating and Improving Multi-task Optimization in Massively Multilingual Models. In *ICLR*.

Yang, S.-W.; Chi, P.-H.; Chuang, Y.-S.; Lai, C.-I. J.; Lakhotia, K.; Lin, Y. Y.; Liu, A. T.; Shi, J.; Chang, X.; Lin, G.-T.; Huang, T.-H.; Tseng, W.-C.; tik Lee, K.; Liu, D.-R.; Huang, Z.; Dong, S.; Li, S.-W.; Watanabe, S.; Mohamed, A.; and yi Lee, H. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Interspeech*, 1194–1198.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. volume 33.