

WAV2VEC-SWITCH: CONTRASTIVE LEARNING FROM ORIGINAL-NOISY SPEECH PAIRS FOR ROBUST SPEECH RECOGNITION

Yiming Wang[†] Jinyu Li[†] Heming Wang^{**} Yao Qian[†] Chengyi Wang[†] Yu Wu[†]

[†] Microsoft Corporation * The Ohio State University, USA

{yimingwang, jinyuli, yaoqian, t-chewang, wu.yu}@microsoft.com, wang.11401@osu.edu

ABSTRACT

The goal of self-supervised learning (SSL) for automatic speech recognition (ASR) is to learn good speech representations from a large amount of unlabeled speech for the downstream ASR task. However, most SSL frameworks do not consider noise robustness which is crucial for real-world applications. In this paper we propose wav2vec-Switch, a method to encode noise robustness into contextualized representations of speech via contrastive learning. Specifically, we feed original-noisy speech pairs simultaneously into the wav2vec 2.0 network. In addition to the existing contrastive learning task, we switch the quantized representations of the original and noisy speech as additional prediction targets of each other. By doing this, it enforces the network to have consistent predictions for the original and noisy speech, thus allows to learn contextualized representation with noise robustness. Our experiments on synthesized and real noisy data show the effectiveness of our method: it achieves 2.9–4.9% relative word error rate (WER) reduction on the synthesized noisy LibriSpeech data without deterioration on the original data, and 5.7% on CHiME-4 real 1-channel noisy data compared to a data augmentation baseline even with a strong language model for decoding. Our results on CHiME-4 can match or even surpass those with well-designed speech enhancement components.

Index Terms— self-supervised learning, contrastive learning, representation learning, robust speech recognition, wav2vec 2.0

1. INTRODUCTION

Training an automatic speech recognition (ASR) system without the need to collect a large amount of transcribed data has been a long-lasting problem, especially for low-resource domains/languages. Previous efforts include model transfer learning, domain adaptation, knowledge distillation/teacher-student learning, and semi-supervised training [1, 2, 3, 4, 5, 6, *inter alia*]. Recently, self-supervised learning (SSL) has emerged as a promising paradigm to tackle this problem. SSL for speech tasks leverages unlabeled data with a self-supervised loss in a pre-training stage, where it is capable of learning good contextualized representations from input speech. Then after fine-tuning the pre-trained model with a small amount of transcribed speech in a conventional supervised manner, the performance can match those trained directly with a much larger amount of labeled data [7, 8, 9, 10, 11, 12, 13]. Existing SSL methods for speech include contrastive predictive coding (CPC) [7, 8], auto-regressive predictive coding [9], and masked predictive encoding [10, 14, 13]. Certain others may fit in more than one categories above [15, 16]. Moreover, a recent work [17] iteratively performed

pseudo-labeling in a more traditional way and fine-tuning on refined pre-trained models to further push the limits of SSL.

Noise robustness is another challenge for ASR in real-world applications [18]. Speech recordings from real-world scenarios usually contain background noise and noise caused by recording imperfection, resulting in deteriorated ASR performance. Prevailing strategies dealing with this challenge are to plug a dedicated enhancement/denoising module into the pipeline of an ASR system as a front-end to suppress the noise, either by training that module separately [19, 20, 21] or jointly [22, 23] with acoustic models. The motivation of joint training is to alleviate the problem that optimizing enhancement objective independently does not necessarily lead to an optimal solution for the ASR task, even if it improves the intelligibility of speech. In either way it will add complexity to the neural network models.

In this paper we focus on strengthening the noise robustness of the pre-trained model during SSL for ASR. Existing work to this end includes PASE+ [24] where a variety of speech transformations were estimated from contaminated speech, and wav2vec-C [25] where a reconstruction module was added on top of the quantized output of the wav2vec 2.0 network [15] and the reconstruction loss was jointly trained with the existing contrastive loss. Same as wav2vec-C, our work, wav2vec-Switch, is also based on wav2vec 2.0. However, instead of a reconstruction loss, we add another contrastive loss as an auxiliary task to achieve noise robustness. Specifically, we batch both the original speech¹ and its noisy version together and feed them to the network. Then in the contrastive learning the quantized targets in each original-noisy pair are switched, so that both the targets are treated as positive in their respective loss calculation. The motivation is that, if we want the contextualized representation robust to noise, the representation of an original speech should also be able to predict the target of its noisy version and vice versa. Different from the prior work, ours does not involve a process of transforming from one representation to another, but enforces the prediction consistency constraint in the contrastive loss without adding any complexity to networks. Experiments on synthesized noisy data and real-world noisy data from the CHiME-4 challenge [19] indicate the efficacy of our approach. In particular, compared to the baseline with data augmentation applied only, we observe 7.1–11.0% relative word error rate (WER) reduction on synthesized noisy speech and 7.8% reduction on real-world noisy speech when decoding without language model (LM), while still maintaining the performance on the original speech. Even in the presence of a strong neural LM, the WER reduction is 2.9–4.9% and 5.7% respectively. Moreover, our results on the CHiME-4 data, with only the unlabeled 960-hour LibriSpeech audio for pre-training, are comparable to, or even better

^{*}Work done during an internship at Microsoft.

¹We refer to “original speech” rather than “clean speech” to avoid any possible confusion, as the original speech in our case is not necessarily clean.

than, other work with a complicated speech enhancement module.

2. MODELS

2.1. Wav2vec 2.0

We recapitulate the wav2vec 2.0 model on which our method is based. Compared to its precedents [8, 14], wav2vec 2.0 combines masked prediction and contrastive learning into a unified model during pre-training. It has a feature encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ with a raw audio waveform $\mathbf{x} \in \mathbb{R}^T$ as its input, and a latent representation $Z = [\mathbf{z}_1, \dots, \mathbf{z}_T]$ with a time-domain down-sampling through a set of convolutional blocks as its output; a context network $g : \mathcal{Z} \mapsto \mathcal{C}$ that takes the masked Z and outputs a contextualized representation \mathbf{c}_t at each masked position t through several blocks of Transformers [26]; and a quantization module $h : \mathcal{Z} \mapsto \mathcal{Q}$ discretizing the unmasked Z to Q from a finite set of codebook via Gumbel Softmax [27] and product quantization [28]. The contrastive loss is applied at each masked position t , discriminating the true quantized representation \mathbf{q}_t (the positive sample) from K distractors $Q_t^- = \{\mathbf{q}_1^-, \dots, \mathbf{q}_K^-\}$ (the negative samples) drawn from other masked positions within the same training example:

$$\mathcal{L}^C(C, Q) = \sum_{t=1}^N \mathcal{L}_t^C(C, Q) / N \quad (1)$$

$$\mathcal{L}_t^C(C, Q) = -\log \frac{\exp(\text{sim}(\mathbf{c}_t, \mathbf{q}_t))}{\sum_{\mathbf{q}^- \in Q_t^-} \exp(\text{sim}(\mathbf{c}_t, \mathbf{q}^-))} \quad (2)$$

where N is the number of masked positions where the loss is computed, and $\text{sim}(\cdot, \cdot)$ is implemented as the *cosine similarity* function. In addition, a diversity loss \mathcal{L}^D encouraging the codebook utilization is implemented as negative perplexity of the Gumbel Softmax output. The total loss to optimize is:

$$\mathcal{L} = \mathcal{L}^C + \alpha \mathcal{L}^D \quad (3)$$

where α is a coefficient. During fine-tuning, the quantization module is discarded and the parameters inside the feature encoder are frozen. All the other network parameters are updated with the CTC loss [29].

2.2. Wav2vec-Switch

In the pre-training stage of wav2vec 2.0, the decision that distractors are only sampled from masked positions within the same training example, rather than from any examples, is critical, or it will hurt the ASR performance [15]. The reason is that sampling only within the same training example can avoid learning features irrelevant to ASR, e.g., speaker or environmental characteristics. However, no mechanisms are designed to achieve noise robustness during the pre-training: when a noisy utterance is given, both the positive and negative samples that the contrastive loss is using contain noise, and there is no explicit way to differentiate the noise from the speech or to learn a contextualized representation invariant to noise. Our intuition is that, if the contextualized representation is robust to noise, the representation of an original/noisy speech should be capable of predicting the target of the noisy/original speech as well. Motivated by this, we propose wav2vec-Switch as follows.

For each mini-batch of the original waveform $X \in \mathbb{R}^{B \times T}$ where B is the batch size, we duplicate X , apply an independently sampled noise to each row (example), and thus have a noisy version of X as \tilde{X} . Then both X and \tilde{X} are fed into the wav2vec 2.0 network in parallel and forwarded through the feature encoder f , the

context network g , and the quantization module h . At this point we have 4 quantities²:

$$\begin{aligned} C &= g(f(X)), & Q &= h(f(X)) \\ \tilde{C} &= g(f(\tilde{X})), & \tilde{Q} &= h(f(\tilde{X})) \end{aligned} \quad (4)$$

In addition to the standard contrastive loss described in Eq. (1) where the loss takes (C, Q) or (\tilde{C}, \tilde{Q}) as its input arguments, we also switch the quantized targets Q and \tilde{Q} , and form two more tuples for the loss: (C, \tilde{Q}) and (\tilde{C}, Q) . Therefore we obtain 4 contrastive loss quantities: $\mathcal{L}^C(C, Q)$, $\mathcal{L}^C(\tilde{C}, \tilde{Q})$, $\mathcal{L}^C(C, \tilde{Q})$, and $\mathcal{L}^C(\tilde{C}, Q)$. The new loss would be:

$$\mathcal{L}_{\text{switch}}^C(C, Q, \tilde{C}, \tilde{Q}) = \mathcal{L}^C(C, Q) + \mathcal{L}^C(\tilde{C}, \tilde{Q}) + \lambda \left(\mathcal{L}^C(C, \tilde{Q}) + \mathcal{L}^C(\tilde{C}, Q) \right) \quad (5)$$

where the coefficient λ controls the weight of the term calculated from the switched targets, relative to the one obtained from the original targets. Fig. 1 also illustrates how the 4 contrastive loss quantities are calculated. Finally \mathcal{L}^D is added to the total loss the same way as in Eq. (3). Note that when $\lambda = 0$, the method reduces to the “wav2vec 2.0 with data augmentation” case, which will be compared against as the baselines in Section 3.

We also need to keep in mind that the network’s internal states for any specific input pair (X, \tilde{X}) should be *identical*. *Identical* means that not only the architecture and parameters but also anything inside the network that relies on random states, including masked positions for the context network and dropout masks of all the dropout layers, should be the same for X and \tilde{X} . Otherwise the representations of the original speech and its noisy version will not follow with each other, and our approach will not behave as what we expect or learn representations with a meaningful interpretation. The ablation study in Section 3.7 also verifies the importance of enforcing such a constraint.

In practice, we batch X and \tilde{X} together, and feed the resulting “large” mini-batch to the network. Every time before a random function is invoked, we save the current random state. That random state is restored immediately after the random function is invoked for the first half of the mini-batch, and then the same function is executed for the second half. By doing in this way, we are able to ensure that the network’s internal states for X and \tilde{X} are always identical.

3. EXPERIMENTS

3.1. Datasets

In order to validate the noise robustness of our proposed approach for ASR, we evaluate it on both synthesized and real-world noisy data. In this section, we first introduce how our data was prepared for these experiments. For the SSL paradigm, there are normally 3 stages: 1) self-supervised pre-training with a large unlabeled dataset; 2) fine-tuning with a relatively small set of labeled data; and 3) testing data in the target domain. We will explain the corresponding setup for each of these 3 stages.

In the synthesized data experiments, both the pre-training and test sets were formed by mixing the LibriSpeech corpus [30] with noise randomly drawn from the MUSAN corpus [31], and the signal-to-noise ratio (SNR) was uniformly sampled from the range between 5 and 10 dB. Note that the MUSAN corpus contains 3 categories

²Here we slightly abuse the use of the notation $C, Q, \tilde{C}, \tilde{Q}$ to denote the batched versions of their respective quantities described in Section 2.1.

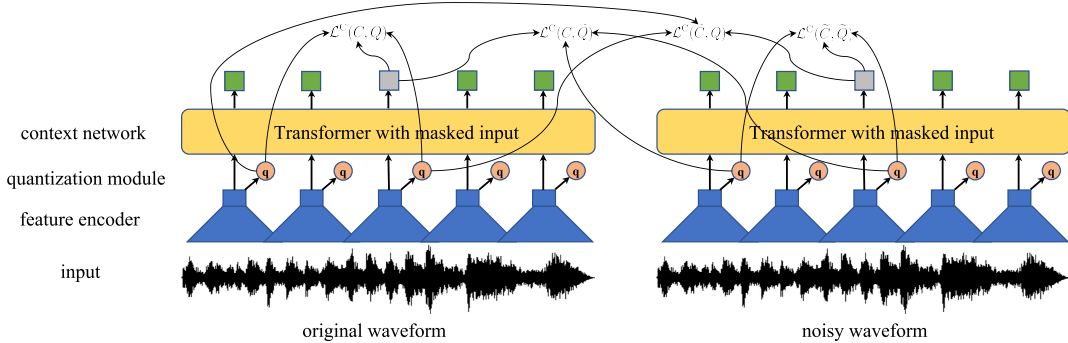


Fig. 1: This figure illustrates how the 4 contrastive losses are calculated in the proposed wav2vec-Switch model. The original-noisy pair of speech is fed into the network with identical internal states simultaneously. The loss quantities $\mathcal{L}^C(C, \tilde{Q})$ and $\mathcal{L}^C(\tilde{C}, Q)$ in the middle are obtained by switching the quantized representations of the original and noisy speech as prediction targets.

of noise: *music*, *noise* and *speech*. To avoid introducing potential confusion with actual speech when learning speech representations, we only used the first 2 categories of the noise from MUSAN and added them on-the-fly to the original 960-hour LibriSpeech training set for pre-training. For the test sets, we prepared the original `test-clean` and `test-other`, and several synthesized versions with different levels of SNR or different subsets of the MUSAN corpus being added, in order to evaluate the model on test sets with a variety of mismatched conditions. We used the original LibriSpeech `train-clean-100` labeled data for fine-tuning, and the reason is that we would like the final model not to degrade its performance on the original sets.

In the real noisy data experiments, we used the data from the CHiME-4 challenge³ [19]. The data in CHiME-4 is based on the “WSJ0” portion of the Wall Street Journal corpus [32] (WSJ), either obtained by recording the actual talker’s speech in real noisy environments (on a bus, cafe, pedestrian area, and street junction) (denoted as “real data”), or artificially mixing the clean WSJ speech with background noise and recording them using a 6-channel distant microphone array and a close-talk microphone (denoted as “simulated data”). Specifically, we chose the real 1-channel track for testing. All the channels but channel 2 from both the “real data” and “simulated data” were used as independent utterances for pre-training and fine-tuning. Channel 2 was excluded because it is the one corresponding to the “behind” microphone which is of low recording quality. Since the CHiME-4 corpus is too small (less than 100 hours) to pre-train from scratch, we continued the pre-training with the CHiME-4 corpus on top of a model already pre-trained with LibriSpeech. The detailed comparisons between with and without continual pre-training are available in Table 3.

3.2. Model Pre-training

Our implementation was made upon the official release of wav2vec 2.0 from FAIRSEQ [33], and the network architecture used throughout this paper was identical to the LibriSpeech BASE configuration specified in [15]: 12 transformer blocks with hidden dimension 768 and 8 heads. Most of the training settings and strategies for the BASE model were also carried over into ours, except that the batch sizes were doubled to accommodate the pairs of original-noisy examples. The coefficient λ in Eq. (5) was set to 0.3 empirically for all the wav2vec-Switch experiments. We also found that a smaller learning

rate (e.g., 1/5 of the one used in the pre-training with LibriSpeech) for the continual training led to better ASR performance.

All the models were trained with 32 NVIDIA Tesla V100 GPUs, each with 32GB memory (required for the double-sized batches). We picked the best checkpoint (in terms of the validation loss), rather than the last one, for continual pre-training or fine-tuning after that, so that we do not need to worry about the risk of over-training.

3.3. Model Fine-tuning

We followed the `base_100h` setup in the wav2vec 2.0 code except the specific data being used for the CHiME-4 experiments. All the fine-tuning models were trained with 2 GPUs. We chose the best checkpoint according to the validation WER for final evaluations.

3.4. Decoding and Language Models

As we employed the CTC loss for fine-tuning, incorporating a language model during decoding is crucial for the best performance. For the LibriSpeech experiments, we downloaded an existing Transformer word-level language model⁴ and adopted the same decoding configurations as those corresponding to the BASE 100-hour experiment introduced in the wav2vec 2.0 paper. For the CHiME-4 experiments, an LSTM-based word-level language model with a vocabulary size of 65,000 was trained on the text portion of the WSJ corpus (see [34] for model details), and then the same decoding strategy was performed. The LM weight was tuned separately for each model.

3.5. Results on Synthesized Noisy Data

We first show the WER (%) results on both the original and synthesized noisy sets under the matched condition in Table 1. Besides wav2vec-Switch (the third row), we also include the results from the wav2vec 2.0 model pre-trained on the original 960-hour LibriSpeech corpus (the first row), and the model trained on the synthesized noisy data (the second row, a.k.a. the data augmentation baseline). It is not surprising that without “seeing” the noisy data in training, the performance on the noisy test sets is much worse than that on the original ones: WERs increase by 2.5–3.2 times. After adding noise to the training data, the performance on the noisy data gets greatly improved, while not changing significantly on the original sets. When further replacing the wav2vec 2.0 model with wav2vec-Switch, the performance on the noisy sets is improved relatively by 7.1–11.0%

³http://spandh.dcs.shef.ac.uk/chime_challenge/CHiME4/index.html

⁴https://dl.fbaipublicfiles.com/wav2letter/sota/2019/lm/lm_librispeech_word_transformer.pt

without LM, or 2.9–4.9% with LM. In addition, the WERs on the original sets are even slightly better than those in the baseline in the “no LM” case, and remain almost the same if the LM is used. These results are promising as it is shown clearly that the model trained with wav2vec-Switch achieves noise robustness on the synthesized noisy data without hurting the performance on the original data.

Table 1: Results on the original and synthesized noisy LibriSpeech test sets under the matched condition.

	LM	Original		Noisy (5–10 dB)	
		test-clean	test-other	test-clean	test-other
wav2vec 2.0	N	5.9	13.4	15.6	33.1
	Y	2.6	6.6	8.0	21.3
+ MUSAN <i>music+noise</i> (5–10 dB) (Baseline)	N	6.1	14.1	8.2	19.8
	Y	2.6	6.7	3.4	10.2
wav2vec-Switch	N	5.8	13.6	7.3	18.4
	Y	2.7	6.7	3.3	9.7

Next we present the results under mismatched conditions. *Mismatched conditions* refers to the cases where the noisy conditions for testing are different from those for training. Given that the noisy condition for training was *music+noise* (5–10 dB), we created 3 versions of noisy test sets with mismatched conditions: 1) *music+noise* (0–5 dB) was the version where the SNR range in the test sets has no overlap with that in the training set; 2) *speech* (5–10 dB) was the one where the noise type being added to the test sets is different; and 3) *speech* (0–5 dB) was when both the SNR range and noise type are different. Table 2 demonstrates the results without LM along with the gains obtained by using the proposed wav2vec-Switch instead of wav2vec 2.0. We can see that the improvements hold in all the mismatched conditions. However, the relative gain becomes smaller when the degree to which the test noise condition mismatch with the training noise condition gets larger.

Table 2: Results on the synthesized noisy sets under different mismatched noisy conditions (without LM).

	<i>music+noise</i> (0–5 dB)		<i>speech</i> (5–10 dB)		<i>speech</i> (0–5 dB)	
	test-clean	test-other	test-clean	test-other	test-clean	test-other
wav2vec 2.0 + MUSAN <i>music+noise</i> (5–10 dB)	11.0	26.1	25.7	52.9	54.7	82.4
wav2vec-Switch	9.7	24.5	23.8	50.4	52.7	80.7
Gain (%)	11.8	6.1	7.4	4.7	3.7	2.1

3.6. Results on Real Noisy Data

We now evaluate wav2vec-Switch on the CHiME-4 corpus. The 1-channel track real noisy data was used for model validation and evaluation. We also include the best results reported in the challenge [35] and other more recent ones [36, 21]. All of them adopted a supervised training paradigm and had a dedicated speech enhancement module to preprocess the noisy speech input before an acoustic modeling module, and some of them even leveraged model ensemble. As presented in Table 3, the best results are from wav2vec-Switch with the continual pre-training. The relative improvement from the corresponding baseline is 7.8% without LM (16.5 vs. 17.9), or 5.7% with LM (6.6 vs. 7.0). It is also worth noting that, without any speech enhancement, our self-supervised approach followed by a simple CTC fine-tuning achieves better results than those using carefully designed enhancement algorithms. The additional data we were using was just the unlabeled 960-hour LibriSpeech audio and the MUSAN corpus.

Table 3: Results on the CHiME-4 real 1-channel dev/eval sets.

	continual pre-training	LM	dev	eval
Chen et al. (Kaldi Baseline) [36] (2018)		Y	5.6	11.4
Du et al. [35] (2016)		Y	4.6	9.2
Wang et al. [21] (2020)		Y	3.5	6.8
wav2vec 2.0 + MUSAN <i>music+noise</i> (5–10 dB) (Baseline)	N	N	10.6	17.6
		Y	3.7	7.2
	Y	N	10.7	17.9
		Y	4.6	7.0
wav2vec-Switch	N	N	10.2	16.8
		Y	3.6	7.1
	Y	N	10.0	16.5
		Y	3.5	6.6

3.7. Ablation Study

To investigate the impact of not keeping the dropout masks or the masked positions the same within each original-noisy speech pair, we conducted one experiment on LibriSpeech where the dropout masks within the context network were different between the examples within each pair, and another experiment where the masked positions were different⁵. We report the results without LM in Table 4. For easy comparisons, we also copy 2 relevant rows from Table 1. It shows that despite a clear degradation from the one with identical dropout masks, the one with non-identical dropout masks is still better than the baseline on all the test sets except the original *test-clean*. However, having different masked positions resulted in a significant deterioration in WER (15.8–27.6% increase), which is expected since we cannot force a representation to still have consistent predictions when they are actually making predictions for different positions before and after the switch, and consequently it will end up with a sub-optimal solution if doing so. These two experiments attest the importance of maintaining identical dropout masks and masked positions for input pairs in wav2vec-Switch.

Table 4: Results on the original and synthesized noisy LibriSpeech sets related to whether to keep identical dropout masks or masked positions between speech pairs. The models were evaluated without LM. Row 1 and 2 are from Table 1 for clearer comparisons.

	Original		Noisy (5–10 dB)	
	test-clean	test-other	test-clean	test-other
wav2vec 2.0 + MUSAN <i>music+noise</i> (5–10 dB) (Baseline)	6.1	14.1	8.2	19.8
wav2vec-Switch w/ identical dropout masks	5.8	13.6	7.3	18.4
wav2vec-Switch w/o identical dropout masks	6.4	13.8	7.8	18.4
wav2vec-Switch w/o identical masked positions	7.4	16.5	8.6	21.3

4. CONCLUSION

We present wav2vec-Switch, a self-supervised learning model based on wav2vec 2.0 that infuses noise robustness into the contrastive representation learning for ASR without introducing extra components to neural networks. Robustness to noise is achieved by feeding pairs of original and noisy speech into the network, and enforcing the prediction consistency for the original and noisy speech, i.e. treating the corresponding quantized targets of the original and noisy speech as positive in the contrastive learning. Experiments on both synthesized and real-world noisy speech exhibit the power of our proposed method for robust ASR. Future work includes pre-training on even larger unlabeled data, exploring other ways of getting contrastive samples, and extending our method beyond the contrastive learning.

⁵We still keep the number of masked positions the same to ensure dimension match after switching the targets.

5. REFERENCES

- [1] Dong Wang and Thomas Fang Zheng, “Transfer learning for speech and language processing,” in *Proc. APSIPA*, 2015.
- [2] Amit Das and Mark Hasegawa-Johnson, “Cross-lingual transfer learning during supervised training in low resource scenarios,” in *Proc. Interspeech*, 2015.
- [3] Taichi Asami, Ryo Masumura, Yoshikazu Yamaguchi, Hirokazu Masataki, and Yushi Aono, “Domain adaptation of DNN acoustic models using knowledge distillation,” in *Proc. ICASSP*, 2017.
- [4] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong, “Large-scale domain adaptation via teacher-student learning,” in *Proc. Interspeech*, 2017.
- [5] Changhan Wang, Juan Miguel Pino, and Jiatao Gu, “Improving cross-lingual transfer learning for end-to-end speech recognition with speech translation,” in *Proc. Interspeech*, 2020.
- [6] Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur, “Semi-supervised training of acoustic models using lattice-free MMI,” in *Proc. ICASSP*, 2018.
- [7] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” in *Proc. NeurIPS*, 2018.
- [8] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in *Proc. Interspeech*, 2019.
- [9] Yu-An Chung and James R. Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *Proc. ICASSP*, 2020.
- [10] Andy T. Liu, Shu-Wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in *Proc. ICASSP*, 2020.
- [11] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *Proc. ICASSP*, 2020.
- [12] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang, “UniSpeech: Unified speech representation learning with labeled and unlabeled data,” in *Proc. ICML*, 2021.
- [13] Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “HuBERT: How much can a bad teacher benefit ASR pre-training?,” in *Proc. ICASSP*, 2021.
- [14] Alexei Baevski, Steffen Schneider, and Michael Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *Proc. ICLR*, 2020.
- [15] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS*, 2020.
- [16] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu, “W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in *Proc. ASRU*, 2021.
- [17] Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” *NeurIPS SAS Workshop*, 2020.
- [18] Jinyu Li, Li Deng, Reinhold Häeb-Umbach, and Yifan Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*, Academic Press, 2015.
- [19] Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech and Language*, vol. 46, pp. 535–557, 2017.
- [20] Keisuke Kinoshita, Tsubasa Ochiai, Marc Delcroix, and Tomohiro Nakatani, “Improving noise robust automatic speech recognition with single-channel time-domain enhancement network,” in *Proc. ICASSP*, 2020.
- [21] Zhong-Qiu Wang, Peidong Wang, and DeLiang Wang, “Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1778–1787, 2020.
- [22] Aswin Shanmugam Subramanian, Xiaofei Wang, Murali Karthick Baskar, Shinji Watanabe, Toru Taniguchi, Dung T. Tran, and Yuya Fujita, “Speech enhancement using end-to-end speech recognition objectives,” in *Proc. WASPAA*, 2019.
- [23] Xuankai Chang, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe, “MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition,” in *Proc. ASRU*, 2019.
- [24] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio, “Multi-task self-supervised learning for robust speech recognition,” in *Proc. ICASSP*, 2020.
- [25] Samik Sadhu, Di He, Che-Wei Huang, Sri Harish Mallidi, Minhua Wu, Ariya Rastrow, Andreas Stolcke, Jasha Droppo, and Roland Maas, “Wav2vec-C: A self-supervised model for speech representation learning,” in *Proc. Interspeech*, 2021.
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Proc. NeurIPS*, 2017.
- [27] Eric Jang, Shixiang Gu, and Ben Poole, “Categorical reparameterization with gumbel-softmax,” in *Proc. ICLR*, 2017.
- [28] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, “Product quantization for nearest neighbor search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011.
- [29] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006.
- [30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015.
- [31] David Snyder, Guoguo Chen, and Daniel Povey, “MUSAN: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.
- [32] Douglas B. Paul and Janet M. Baker, “The design for the wall street journal-based CSR corpus,” in *Proc. ICSLP*, 1992.
- [33] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proc. NAACL-HLT: Demonstrations*, 2019.
- [34] Yiming Wang, Tongfei Chen, Hainan Xu, Shuoyang Ding, Hang Lv, Yiwen Shao, Nanyun Peng, Lei Xie, Shinji Watanabe, and Sanjeev Khudanpur, “Espresso: A fast end-to-end neural speech recognition toolkit,” in *Proc. ASRU*, 2019.
- [35] Jun Du, Yan-Hui Tu, Lei Sun, Feng Ma, Hai-Kun Wang, Jia Pan, Cong Liu, Jing-Dong Chen, and Chin-Hui Lee, “The USTC-iFlytek system for CHiME-4 challenge,” in *Proc. CHiME-4 Challenge*, 2016.
- [36] Szu-Jui Chen, Aswin Shanmugam Subramanian, Hainan Xu, and Shinji Watanabe, “Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline,” in *Proc. Interspeech*, 2018.