



SPLICEOUT: A Simple and Efficient Audio Augmentation Method

Arjit Jain¹, Pranay Reddy Samala¹, Deepak Mittal², Preethi Jyothi¹, Maneesh Singh²

¹Indian Institute of Technology Bombay

²Verisk Analytics

arjit@cse.iitb.ac.in, pranayr@cse.iitb.ac.in, deepak.mittal@verisk.com,
pjyothi@cse.iitb.ac.in, maneesh.singh@verisk.com

Abstract

Time masking has become a *de facto* augmentation technique for speech and audio tasks, including automatic speech recognition (ASR) and audio classification, most notably as a part of SpecAugment. In this work, we propose SPLICEOUT, a simple modification to time masking which makes it computationally more efficient. SPLICEOUT performs comparably to (and sometimes outperforms) SpecAugment on a wide variety of speech and audio tasks, including ASR for seven different languages using varying amounts of training data, as well as on speech translation, sound and music classification, thus establishing itself as a broadly applicable audio augmentation method. SPLICEOUT also provides additional gains when used in conjunction with other augmentation techniques. Apart from the fully-supervised setting, we also demonstrate that SPLICEOUT can complement unsupervised representation learning with performance gains in the semi-supervised and self-supervised settings.

Index Terms: speech recognition, data augmentation, audio classification

1. Introduction

SpecAugment [1] offered a simple alternative of directly transforming an audio spectrogram (i.e. a visual representation of the audio signal) by randomly warping across the time axis or masking consecutive chunks of time (i.e. *time masking*) or frequency channels (i.e. *frequency masking*) in the spectrogram. It was shown to be very effective for ASR, and has since been used for audio classification tasks as well [2, 3, 4]. Investigations into new augmentation techniques that work broadly across different audio and speech tasks have been fairly limited. This could be partly attributed to the opaque nature of the audio spectrogram. Unlike image transformations (e.g., translation, shear, brightness, etc.) or text transformations (e.g., synonym replacements, word swapping, etc.) whose effects on a training instance can be easily visualized and understood, transformations on audio spectrograms are harder to interpret, and thus more challenging to define.

In this work, we propose SPLICEOUT, a simple modification to time masking that makes it more computationally efficient. Instead of masking splices of consecutive time-steps in the audio input, SPLICEOUT operates by entirely removing these time slices from the audio input and stitching the remaining parts together. Figure 1 illustrates the main difference between SPLICEOUT and time masking.

While augmentation techniques in prior work have been typically proposed for specific target tasks (e.g. SpecAugment for ASR), we present SPLICEOUT as a broadly applicable augmentation technique for most sequence labeling or multi-class classification tasks using speech or audio as input. We

substantiate the claim that SPLICEOUT has broad applicability by showing that it performs as well as (and at times better than) time masking on many speech and audio tasks including ASR, speech translation, sound and music classification. (SPLICEOUT can be applied to both raw waveform and spectrogram-based input formats.) We demonstrate the effectiveness of SPLICEOUT in both low and high resource training regimes and show ASR results for a number of different languages. Besides fully supervised tasks, we also demonstrate that SPLICEOUT can benefit representation learning by showing improvements in semi-supervised and self-supervised settings. Along with establishing that SPLICEOUT is useful as a standalone technique, we also show that SPLICEOUT can be complementary to time masking. It is noteworthy that in all our experiments SPLICEOUT was used as a drop-in replacement for SpecAugment whose hyperparameters were presumably tuned for the task at hand; any performance improvements we derived with SPLICEOUT were without using any separate hyperparameter tuning.

2. Related Work

Masking-based augmentation (and regularization) techniques have been used extensively in machine learning, from masking weights [5] and hidden states [6, 7, 8, 9] in neural networks, to masking pixels in images [10, 11, 12, 13], words in text [14, 15, 16], and time-steps in audio [1, 17] and time-series data. While masking has been employed as a commonly used augmentation technique, it has also been used in representation learning where the objective involves learning to reconstruct masked out information [15, 13, 14]; this reconstruction objective forms the basis of most of the recent semi-supervised and self-supervised learning approaches. Masking-based augmentation techniques can also be used with contrastive losses for representation learning (that have seen a recent resurgence [18, 19, 20] where different augmented views of the same data samples can serve as positive examples.

SPLICEOUT would qualify as a time masking technique, except we delete blocks of consecutive time-steps instead of masking them. In subsequent experimental comparisons, we use SPLICEOUT as a drop-in replacement for time masking. SPLICEOUT is also complementary to other augmentation techniques as will be demonstrated in our Experiments.

3. SPLICEOUT

SPLICEOUT is parameterized by N , the number of time intervals to splice, and T , the maximum width of a time interval. Given a log-mel spectrogram with τ time steps, SPLICEOUT selects N intervals, each of the form $[t_0, t_0 + t)$ where $t \in \mathbb{Z}^{0+}$ is sampled from the uniform distribution $U(0, T)$, and $t_0 \in \mathbb{Z}^{0+}$ is chosen from $[0, \tau - t)$. The input spectrogram is modified by

removing all time steps in the union of the selected intervals.

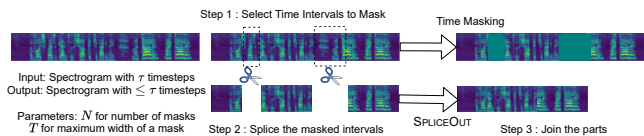


Figure 1: *Illustration of SPLICEOUT and time masking.*

Figure 1 illustrates how SPLICEOUT decreases the length of the output spectrogram, as opposed to Time-Masking which results in an output spectrogram of the same length as the original input spectrogram. As we increase the amount of masking, SPLICEOUT deletes larger numbers of time intervals from the input data, and the resulting shorter inputs in turn reduce the memory footprint, and time, required for training. Time-Masking, on the other hand, does not gain any computational advantage with respect to the memory or the time required for training, with increased amount of masking.

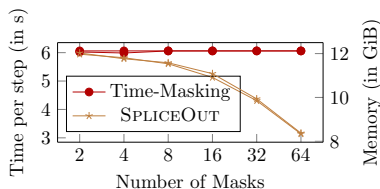


Figure 2: *Comparison of running time and memory requirements during training using Time-Masking and SPLICEOUT augmentations, with varying number of masks.*

We empirically verify this for an ASR task on the LibriSpeech 100-hr benchmark. Figure 2 plots the time taken per training step, and the memory required for training, for both Time-Masking and SPLICEOUT, for different values of N , while keeping the maximum width T constant at 40 time steps. As expected, we observe that SPLICEOUT is indeed more time-efficient and memory-efficient than Time-Masking. For a fixed masking budget, we see that SPLICEOUT is more computationally efficient than Time-Masking, with larger gains as we increase the amount of masking.

4. Experiments

We demonstrate the effectiveness of SPLICEOUT on a wide variety of speech and audio tasks spanning different model architectures, learning paradigms and toolkits.

4.1. Speech-based Sequence Labeling Tasks

4.1.1. ASR: LibriSpeech

LibriSpeech is a widely used ASR benchmark consisting of English read speech. We use the clean 100-hr subset for training and evaluate on the standard dev (clean/other) and test (clean/other) sets. The “other” evaluation datasets comprise speech samples that are more acoustically challenging. Our base model is the large variant of the Conformer model [21], Conformer(L), which is a state-of-the-art network for ASR and is implemented using the ESPnet toolkit [22]. We use 83-dimensional log mel-filterbank + pitch features as input. Each model is trained for 60 epochs, and model averaging is performed on weights from 5 epochs with the best performance on

the validation set. Speed perturbation, with speeds 0.9, 1.0, 1.1, is applied to the training data. Beam search is used during decoding, without invoking any external language model.

Table 1: *WERs on LibriSpeech test sets, using TM, FM and SPLICEOUT (SO), with $N = 2$.*

Augmentation	test-clean	test-other
TM	7.6±0.19	21.8±0.31
SO	7.5±0.18	21.4±0.31
FM + TM	7.2±0.17	18.3±0.28
FM + SO	7.2±0.18	18.2±0.29
TW + FM + TM	7±0.16	18.1±0.28
TW + FM + SO	7.1±0.17	17.9±0.29
TW + FM + TM + SO	7.1±0.18	17.7±0.29

Table 2: *Effect of increasing the number of masks, N , in TM and SO augmentations, on WERs of LibriSpeech test sets.*

N	Method	test-clean	test-other
2	TM	7.6±0.19	21.8±0.31
	SO	7.5±0.18	21.4±0.31
4	TM	7.3±0.18	20.4±0.30
	SO	7±0.16	20.3±0.31
8	TM	6.8±0.17	19.2±0.29
	SO	6.8±0.18	19±0.30

Table 1 shows the word error rates (WERs) using SPLICEOUT as a replacement for Time-Masking (TM), both with and without the presence of other augmentations like time warping (TW) and frequency masking (FM). The time warp parameter for TW is set to 5. We use 2 masks of maximum width 30 for FM. For TM and SO, $N = 2$ and $T = 40$. We observe that SPLICEOUT either performs comparably or a bit better when used as a replacement for TM. Interestingly, using SPLICEOUT in conjunction with SpecAugment (TW, FM, TM) yields further improvements in performance on test-other. Table 2 shows how WERs change with increasing N for TM and SO. For a fixed N , we observe that SPLICEOUT almost consistently outperforms TM. (With larger values of $N \geq 16$, the performance starts to degrade.)

Semantic Mask. While SpecAugment randomly masks time intervals in a spectrogram, Semantic-Mask [23] proposes masking out regions which correspond to a word or word-piece in the transcription, thus encouraging the decoder to learn a better internal language model. Similar to how we modify TM to implement SPLICEOUT, we modify Semantic-Mask to implement Semantic-Splice wherein we splice, or delete, certain intervals which correspond to a word or word-piece. We compare Semantic-Mask and Semantic-Splice on the LibriSpeech 100-hr dataset. We follow the experimental setup described in [23]. 15% of the tokens are masked or spliced out in Semantic-Mask and Semantic-Splice, respectively. A transformer model is used as the base architecture, with convolutional layers for encoding, instead of positional encodings, similar to [23]. Speed perturbation, with speeds 0.9, 1.0, 1.1 is used, along with SpecAugment. Models for both Semantic-Mask and Semantic-Splice

were trained for 120 epochs, and the model with the best validation performance was used for testing. Table 3 reports WERs for both methods showing that Semantic-Splice consistently outperforms Semantic-Mask on both the test sets.

Table 3: WERs on LibriSpeech test sets comparing Semantic-Mask and Semantic-Splice.

Method	test-clean	test-other
Semantic-Mask	9.2	22
Semantic-Splice (Ours)	8.8	21.5

4.1.2. ASR for Multiple Languages: CommonVoice

The CommonVoice corpus [24] is a multilingual corpus of read speech in 38 languages. We train ASR models for eleven different languages spanning training sizes ranging from 10 hours to 116 hours. We use a conformer architecture adapted for ESPnet [25]. The same model architecture and hyperparameter settings are used for all languages. We train the model for 50 epochs with 83-dimensional features (80 log-mel filterbank and 3 pitch) and byte pair encoding (BPE)-based encoded transcripts (with a vocabulary of size 150). Speed perturbation is applied, with speeds 0.9, 1.0 and 1.1, during training. Beam search decoding without external language model is used during inference. Parameter values for TW, FM, TM are the same as the previous section.

Table 4 shows WERs for all eleven languages. Our experiments demonstrate that SPLICEOUT consistently outperforms TM, and obtains statistically significant WER reductions on four low-resource languages, Kyrgyz, Swedish, Tatar and Ukrainian (at $p < 0.05$ using the MAPSSWE test preferred for ASR evaluations [26]).

4.1.3. Speech Translation: Libri-Trans

Libri-Trans is a speech translation (ST) benchmark with approximately 240h of English read speech (from Librispeech) aligned with French text [27]. The base model for ST consists of an ASR encoder and a machine translation (MT) decoder. We initialized the encoder with a SpecAugment-pretrained ASR Transformer and the decoder with a pretrained MT Transformer. Both the models employ BPE units with a vocabulary of size 1K and use joint source and target vocabularies. Speed perturbation, with speeds 0.9, 1.0 and 1.1, was used for ST training. Finetuning is performed for 50 epochs, and model averaging is performed on weights from the 5 epochs with best validation accuracy for evaluation. The ST experiments were conducted using the ESPNet-ST framework [28]. Table 5 shows marginal improvements in BLEU scores on the dev and test sets with using SPLICEOUT as opposed to TM.

4.2. Audio Classification Tasks

We compare SPLICEOUT with Time-Masking on audio classification Tasks. Environmental Sound Classification (ESC-50) [29] and UrbanSound8K [3] are two well-established benchmarks in sound classification containing 50 and 10 sound classes, respectively. State-of-the-art techniques on these datasets rely on transfer learning, where a network pretrained on a large audio classification dataset like AudioSet [30] is further fine-tuned on labeled data. We use the CNN10

architecture [2] that is pretrained on AudioSet and takes mel-spectrograms as input. The experimental setup is similar to [3].

Along with TM, and/or SO, Mixup (MX) and Frequency Masking (FM) are used for data augmentation. We use 2 masks for FM, with a maximum frequency width of 8. The α parameter for mixup is set to 1. For TM and SO, $N = 2$ and $T = 24$.

Table 6 shows that SPLICEOUT consistently improves performance on all three metrics for ESC-50, Accuracy, $F1_{\text{micro}}$ and mean AP compared to TM, and performs as well as TM on UrbanSound8K.

4.3. Representation Learning: CLAR

Contrastive Learning of Audio Representation (CLAR) [17] builds on SimCLR [19] to provide a framework for semi-supervised and self-supervised representation learning for audio data using multiple augmentations. We use the same experimental setup as CLAR, i.e. we perform our experiments on the Speech Commands-10 dataset consisting of speech samples corresponding to 10 isolated-word commands. ResNet18 is the base encoder model with the output dimension set to 512. The projection head used to generate \mathbf{z} is implemented as three fully-connected layers with ReLU activations. To evaluate the learned representations, a linear classifier is trained on the frozen features, as has been done in prior work [31, 32, 18, 19]. The batch size is set to 512, and global batch normalization is used.

Table 7 compares TM with SO when used in conjunction with FD for both semi-supervised and self-supervised representation learning. In 3 out of 4 labeled data settings, SPLICEOUT yields performance improvements compared to TM.

5. Discussion

We present a comprehensive empirical evaluation of SPLICEOUT as a generic audio augmentation technique. We show that it works with different types of audio signals (speech in different languages, sounds), different tasks (ASR, speech translation, classification), different model architectures and optimization functions (encoder-decoder conformer models, CNN10/CNN14 convolutional networks), different learning paradigms (fully supervised, semi-supervised, self-supervised), and different toolkits (ESPnet, speechbrain). We emphasize that all our results were obtained *without any* hyperparameter tuning. We used SPLICEOUT as a drop-in replacement for SpecAugment in the existing implementations. Hyperparameters of the latter were presumably tuned to work well for the specific target tasks. Our results show that, even without any tuning, SPLICEOUT is at least as good as (and sometimes better than) Time-Masking.

Properties of SPLICEOUT. It has been argued that a data augmentation technique (as opposed to a regularization technique) should result in data that is “consistent with observations that may be seen by the model” [34]. Motivated by this, in addition to the empirical evaluation of SPLICEOUT in terms of accuracy in various tasks, we briefly explore its effects in terms of statistical and perceptual distortion, vis-a-vis Time-Masking.

Firstly, we consider simple time-averaged statistics – specifically, mean and variance – of the modified spectrograms. This is motivated in part by normalization techniques like BatchNorm, and by work on auditory perception that provides evidence that the human auditory system uses time-averaged statistics for input summarization [35, 36]. We provide an empirical comparison of mean and variance of the spectrogram ob-

Table 4: Evaluation of TM and SPLICEOUT, when used with TW and FM, using test WERs across multiple different languages, including several low-resource settings

Augs. TW + FM	Swedish 10 hrs	Turkish 22 hrs	Kyrgyz 22 hrs	Ukrainian 25 hrs	Estonian 27 hrs	Tatar 28 hrs	Czech 29 hrs	Dutch 45 hrs	Portugese 53 hrs	Welsh 96 hrs	Russian 116 hrs
+ TM	33.2	6.7	37.3	14.6	41.2	36.3	16.3	2.1	10.5	15.4	9.4
+ SO	32.1	6.5	36.2	14.0	40.8	35.5	15.7	2.0	10.1	14.8	9.0

Table 5: Evaluating TM and SO using development and test set BLEU scores for the Libri-Trans task. Higher is better.

Augmentation	Dev BLEU	Test BLEU
TW + FM + TM	18.43	17.18
TW + FM + SO	18.57	17.15
TW + FM + TM + SO	18.42	17.35

Table 6: Evaluating TM and SO on two sound classification tasks, with the standard augmentation combinations.

Augmentation	Accuracy	F1 _{micro}	mAP
Dataset: ESC-50			
MX + FM + TM	90.40±0.02	89.42±0.02	94.98±0.01
MX + FM + SO	90.95±0.02	89.96±0.02	95.17±0.01
Dataset: UrbanSound8k			
MX + FM + TM	86.39±0.04	86.32 ±0.04	93.04 ±0.03
MX + FM + SO	86.67 ±0.04	86.31 ±0.04	93.04 ±0.03

tained after applying TM or SPLICEOUT augmentations with that of the original signal. In our comparisons, we also include an additional variant, TM (Mean), which explicitly corrects for the mean by mean imputation i.e., setting the masked parts to the mean of the input spectrogram instead of zero as in TM (Zero). As Figure 3 shows, SPLICEOUT maintains mean and variance better than TM (Zero); it matches TM (Mean), and provides a better match for variance.

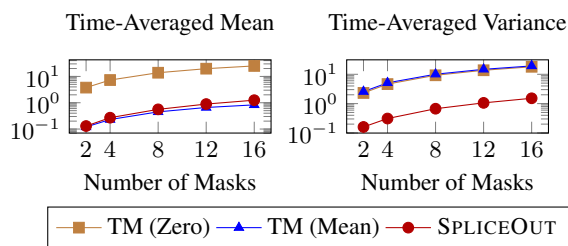


Figure 3: Comparison of % distortion in the Time-Averaged Statistics of different augmentation methods, compared to the unaltered input, with varying number of masks.

To evaluate perceptual distortion, we use two objective intelligibility and quality measures: Perceptual Evaluation of Speech Quality (PESQ) [37] is one of the most widely used metrics that correlates with mean opinion scores from human evaluations of speech signals. Speech-to-Reverberation Modulation energy Ratio (SRMR) [38] is a more recent metric that was proposed in the context of dereverberated speech. Ta-

Table 7: Comparing classification accuracies using SO and TM in semi-supervised (with different amounts of labeled data) and self-supervised settings on the Speech Commands Dataset.

Type	Method	Labeled Data Percentage			
		100%	20%	10%	1%
Sup.	Cross Entropy	94.9	86.4	68.4	28.6
Semi-Sup.	SupCon [33]	96.0	87.9	82.1	26.6
	CLAR (TM)	97.2	94.7	91.7	72.8
	CLAR (SO)	97.4	95.6	92.6	71.2
Self-Sup.	SimCLR (TM)	89.0			
	SimCLR (SO)	88.9			

ble 8 shows PESQ (using both narrowband 8kHz and wideband 16kHz versions) and SRMR values computed for waveforms reconstructed from 100 augmented spectrograms each for 100 random samples from Librispeech using SPLICEOUT and TM (zero and mean imputation). SPLICEOUT yields consistently higher PESQ scores compared to both TM approaches (& performs at par on SRMR). Thus, in the metrics explored, SPLICEOUT better approximates real-life data, and better fits the notion of a data augmentation technique compared to TM, without incurring any efficiency penalties.

Table 8: Perceptual speech metrics, both absolute and relative, comparing the quality of speech modified by different transformations. Higher is better.

Augmentation	Absolute	Relative: PESQ	
	SRMR	Wide-Band	Narrow-Band
TM (Zero)	9.24	3.07	3.35
TM (Mean)	9.14	3.05	3.46
SPLICEOUT	9.24	3.33	3.59

6. Conclusions

We propose SPLICEOUT, a new audio augmentation technique that is a simple modification to time masking and that works well for a variety of audio and speech classification tasks. SPLICEOUT was shown to significantly outperform time masking in low-resource ASR tasks across many languages. SPLICEOUT also offers potential efficiency gains by tuning the number of masks applied to the audio input, unlike time masking where the computational effort is invariant to the amount of masking. We also present analyses to support the claim that SPLICEOUT is better at approximating valid speech samples compared to time masking and hence is a better motivated data augmentation technique.

7. References

- [1] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech*, 2019.
- [2] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," 2020.
- [3] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM-MM*, 2014.
- [4] S. Amiriparian, T. Hübner, M. Gerczuk, S. Ottl, and B. W. Schuller, "Deepspectrumlite: A power-efficient transfer learning framework for embedded speech and audio processing from decentralised data," 2021.
- [5] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," in *ICML 2015*, ser. PMLR, vol. 37, 2015.
- [6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 56, 2014.
- [7] H. Pham and Q. V. Le, "Autodropout: Learning dropout patterns to regularize deep networks," *arXiv preprint arXiv:2101.01761*, 2021.
- [8] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: a regularization method for convolutional networks," in *NeurIPS*, 2018.
- [9] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *ICML*, ser. PMLR, vol. 28, no. 3, 2013.
- [10] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
- [11] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *AAAI*, vol. 34, no. 07, 2020.
- [12] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *CVPR*, 2017.
- [13] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH*, 2000.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [15] C. Donahue, M. Lee, and P. Liang, "Enabling language models to fill in the blanks," in *ACL*, 2020.
- [16] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *NeurIPS*, vol. 29, 2016.
- [17] H. Al-Tahan and Y. Mohsenzadeh, "Clar: Contrastive learning of auditory representations," in *AISTATS*, ser. PMLR, vol. 130, 2021.
- [18] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *arXiv preprint arXiv:2103.03230*, 2021.
- [19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, ser. PMLR, vol. 119, 2020.
- [20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.
- [21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020.
- [22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018.
- [23] C. Wang, Y. Wu, Y. Du, J. Li, S. Liu, L. Lu, S. Ren, G. Ye, S. Zhao, and M. Zhou, "Semantic Mask for Transformer Based End-to-End Speech Recognition," in *Interspeech*, 2020.
- [24] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *LREC*, 2020.
- [25] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP*, 2021.
- [26] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *ICASSP*. IEEE, 1989.
- [27] A. C. Kocabiyyikoglu, L. Besacier, and O. Kraif, "Augmenting librispeech with French translations: A multimodal corpus for direct speech translation evaluation," in *LREC*, 2018.
- [28] H. Inaguma, S. Kiyono, K. Duh, S. Karita, N. Yalta, T. Hayashi, and S. Watanabe, "ESPnet-ST: All-in-one speech translation toolkit," in *ACL*, 2020.
- [29] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *ACM-MM*, 2015.
- [30] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.
- [31] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *NeurIPS*, vol. 33, 2020.
- [32] X. Chen and K. He, "Exploring simple siamese representation learning," *arXiv preprint arXiv:2011.10566*, 2020.
- [33] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *NeurIPS*, vol. 33, 2020.
- [34] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP*, 2020.
- [35] J. H. McDermott, M. Schemitsch, and E. P. Simoncelli, "Summary statistics in auditory perception," *Nature neuroscience*, vol. 16, no. 4, 2013.
- [36] E. A. Piazza, T. D. Sweeny, D. Wessel, M. A. Silver, and D. Whitney, "Humans use summary statistics to perceive auditory sequences," *Psychological Science*, vol. 24, no. 8, 2013.
- [37] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *ICASSP*, vol. 2, 2001.
- [38] T. H. Falk, C. Zheng, and W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, 2010.