INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Learning Emotional Representations from Imbalanced Speech Data for Speech Emotion Recognition and Emotional Text-to-Speech

Shijun Wang[1], *Jón Guðnason*[2], *Damian Borth*[1]

[1]University of St.Gallen, Switzerland
[2]Reykjavik University, Iceland

shijun.wang@unisg.ch, jg@ru.is, damian.borth@unisg.ch

Abstract

Effective speech emotional representations play a key role in Speech Emotion Recognition (SER) and Emotional Text-To-Speech (TTS) tasks. However, emotional speech samples are more difficult and expensive to acquire compared with Neutral style speech, which causes one issue that most related works unfortunately neglect: imbalanced datasets. Models might overfit to the majority Neutral class and fail to produce robust and effective emotional representations. In this paper, we propose an Emotion Extractor to address this issue. We use augmentation approaches to train the model and enable it to extract effective and generalizable emotional representations from imbalanced datasets. Our empirical results show that (1) for the SER task, the proposed Emotion Extractor surpasses the state-of-the-art baseline on three imbalanced datasets; (2) the produced representations from our Emotion Extractor benefit the TTS model, and enable it to synthesize more expressive speech.

Index Terms: emotional representation learning, speech emotion recognition, emotional TTS

1. Introduction

Speech Emotion Recognition (SER) and Emotional Text-To-Speech (TTS) are two important speech-processing tasks. SER studies how to recognize emotional information from speech, while Emotional TTS studies how to generate speech with rich emotional styles given texts. Interest in both tasks is on the rise because they enable the recent machine learning models to better understand and mimic human emotions. The success of these two tasks shares one key role: the effectiveness of speech emotion representations [1, 2]. Specifically, emotional information carried by meaningful representations benefits SER, meanwhile, it renders expressive emotional speech production.

However, one challenge for SER and Emotional TTS is the lack of data, which has not been taken into account in most related approaches [3, 4, 5, 6, 7]. In reality, emotional speech data is more difficult and expensive to acquire compared to Neutral style speech, which results in a common issue that the speech data distributions are highly skewed toward the Neutral emotion class. Therefore, when trying to extract emotional representations, we need to account for imbalanced datasets.

Data augmentation is a common strategy to alleviate the imbalanced data issue, by reducing the bias towards the majority class and increasing the robustness of the model. The authors from [8, 9, 10] apply GANs to augment speech and improve the SER performance on imbalanced datasets. In order to increase the performance of TTS models on small datasets, recent studies propose to augment small corpora with synthesized data, either via voice conversion [11, 12] or a large text-to-speech system [13]. However, most of these TTS works focus on synthesizing augmented speech, rather than using augmentation to achieve more effective speech emotional representations. Meaningful emotional representation is crucial for expressive emotional TTS and can be utilized to affect speech synthesis effectively. The authors in [14] learn style tokens to represent emotional information from reference speech, then these tokens are used to transfer the emotion to the synthesized speech. RF-Tacotron [15] is an extended work of [14]. Instead of using one single token, RFTacotron applies a sequence of vectors to reflect the emotional style of the reference. Such representations provide fine-grained and robust emotional-style control on the synthesized speech. However, it might be difficult for these models to learn effective emotional representations from imbalanced data, because the models might overfit to Neutral emotion, which makes the generated speech less expressive.

Mixup [16] is a popular way to address imbalanced data issues. It augments data by linearly interpolating input samples and the corresponding labels. The effectiveness of Mixup has been widely demonstrated. In [17], the authors use Mixup to improve speech recognition performance on small datasets. Mixup is also applied in [18] to enhance model generalization for speaker verification tasks. Despite the success of Mixup augmentation directly on speech (linear interpolation of two raw speech samples, i.e. raw-level Mixup), the authors in [19] argue that samples created by Mixup are likely to be very unrealistic in terms of speech production, and the effect could be limited due to its linear nature. As a result, [19] applies Mixup to latent space rather than raw inputs, i.e. Mixup on latent representations. Such latent-level Mixup has been explored in other domains as well [20, 21].

In this work, we propose an augmentation approach that can effectively learn speech emotion representations from imbalanced data. Such representations can be used for both SER and Emotional TTS tasks. Specifically, we combine both raw- and latent-level Mixup to gain benefits from both sides. In raw-level Mixup, the model is exposed to a wider variety of input data, and learns to identify the underlying structure in the data, which will likely lead to a more robust and generalizable feature space. On the other hand, Mixup on latent space is exposed to wider information at the level of intermediate activations, which results in a more expressive feature space. Consequently, such manipulation allows us to extract more complex and abstract features of the data, which can be especially beneficial for tasks requiring high-level understanding, To sum up, the combination of these two types of Mixup augmentations has the potential to lead to robust and effective emotional representations.

Additionally, since we can consider raw- and latent-level Mixup two types of augmentation, we force the emotional representations from these two augmentations to attract each other. This is motivated by recent successful Contrastive Learning

10.21437/Interspeech.2023-1595

methods [22, 23, 24, 25]. By forcing the two representations to be similar, the model is encouraged to produce consistent representations for the mixed instances, regardless of whether the mixing is performed at the raw level or at the latent level. Moreover, this consistency might prevent the model from being overly dependent on specific values or features that are likely to be different between the training and test sets in imbalanced datasets. One thing that is worth mentioning is that although we do not feed negative samples as most of the Contrastive Learning methods, we can still learn discriminative representations, since the standard classification loss (cross-entropy) is still applied with the help of emotion labels.

We summarize our contributions as: (1) we propose an augmentation approach that can learn effective emotion representations from imbalanced datasets. (2) The learned representations can be applied to both SER and Emotional TTS tasks. (3) Our experimental results demonstrate that our SER and Emotional TTS models benefit from the learned representations and outperform state-of-the-art models.

2. Methodology

This section describes how we learn emotional representation from speech data and how they are used for SER and Emotional TTS.

2.1. Emotional Representation Learning

We train an Emotion Extractor to extract emotional representations. The pipeline for the training is shown in Fig. 1. We have two speech samples $\mathbf{X}_1$ and $\mathbf{X}_2$ as inputs. The left is raw-level Mixup (standard Mixup). We perform this augmentation on the pair $(\mathbf{X}_1, \mathbf{X}_2)$ to get the mixture $\mathbf{X}_{Rmix}$:

$$\mathbf{X}_{Rmix} = \lambda \mathbf{X}_1 + (1-\lambda)\mathbf{X}_2, \tag{1}$$

where λ is sampled from Beta distribution $Beta(1, 1)$.

Raw-level mixed output $\mathbf{X}_{Rmix}$ is then sent to our Emotion Extractor to produce emotion representation $\boldsymbol{I}_{Rmix}$. Afterward, we average the sequence $\boldsymbol{I}_{Rmix}$ into a vector h_{Rmix}. The standard Mixup loss is further applied to it:

$$\mathcal{L}_{Rmix} = \lambda \text{CE}(h_{Rmix}, y_1) + (1-\lambda)\text{CE}(h_{Rmix}, y_2), \tag{2}$$

and CE$(\cdot, \cdot)$ represents Cross Entropy loss, y_1 indicates the emotion category label of $\mathbf{X}_1$, while y_2 indicates the label of $\mathbf{X}_2$.

On the right side, we perform latent-level Mixup augmentation. The same speech inputs $\mathbf{X}_1$ and $\mathbf{X}_2$ are first independently fed to our Emotion Extractor to produce individual emotion representations $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. Both representations are averaged into h_1 and h_2. Latent-level Mixup is then applied to them:

$$h_{Lmix} = \lambda h_1 + (1-\lambda)h_2. \tag{3}$$

Similar to $\mathcal{L}_{Rmix}$, we apply latent-level Mixup loss on h_{Lmix}:

$$\mathcal{L}_{Lmix} = \lambda \text{CE}(h_{Lmix}, y_1) + (1-\lambda)\text{CE}(h_{Lmix}, y_2). \tag{4}$$

Furthermore, as we mentioned before, in order to encourage the model to produce generalizable emotional representations better, we force the outputs of the raw- and latent-level Mixup augmentations to be similar. Therefore, we employ an extra loss $\mathcal{L}_{sim}$ on h_{Rmix} and h_{Lmix},

$$\mathcal{L}_{sim} = -\ \text{dot}(h_{Rmix}, h_{LMix}), \tag{5}$$

where dot$(\cdot, \cdot)$ indicate the dot product between two vectors.

Finally, we train our Representation Extractor with the total loss:

$$\mathcal{L}_{total} = \mathcal{L}_{Rmix} + \mathcal{L}_{Lmix} + \mathcal{L}_{sim}. \tag{6}$$

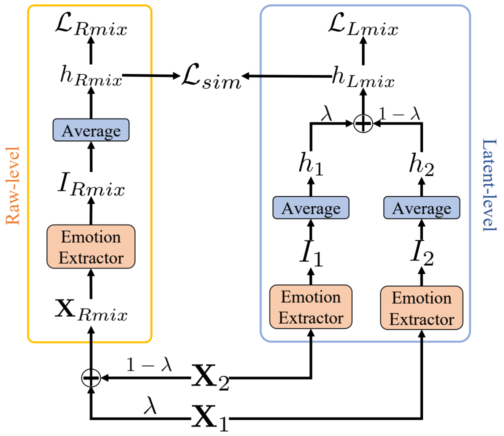

Figure 1: *The training of the Emotion Extractor. $\mathbf{X}_1$ and $\mathbf{X}_2$ are two speech samples. The left path is raw-level Mixup. The mixture $\mathbf{X}_{Rmix}$ is sent to the Emotion Extractor to produce the emotion representation $\boldsymbol{I}_{Rmix}$. h_{Rmix} is an averaged vector. The right path is latent-level Mixup, we first get individual representations $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ from $\mathbf{X}_1$ and $\mathbf{X}_2$. After average, we perform Mixup on vectors h_1 and h_2 to acquire the mixture h_{Lmix}. Finally, raw- and latent-level losses $\mathcal{L}_{Rmix}$ and $\mathcal{L}_{Lmix}$ are two weighted cross entropy losses, while $\mathcal{L}_{sim}$ is to encourage h_{Rmix} and h_{Lmix} to be similar.*

2.2. SER and Emotional TTS

We use two different architectures of the Emotional Encoder for the SER and Emotional Recognition tasks, since we want to make a fair comparison between our models and baselines from different tasks.

2.2.1. SER Task

Since our Emotion Extractor can be directly used for the SER task (Fig. 1). We follow [10] and use VGG19 [26] as our Emotion Extractor for the SER task. To adapt VGG19, we slightly modify our training pipeline in Fig. 1. Specifically, we remove all Average operations since VGG19 gradually downsamples the input via Maxpooling. VGG19 has multiple Convolutional layers and two fully-connected layers, thus, we refer to the last Convolutional layer's output as emotional representation $\boldsymbol{I}$, and the last fully-connected layer's output as vector h.

2.2.2. Emotional TTS Task

We use RFTacotron [15] as our backbone TTS to convert the phonemes to speech, given emotional representations. RFTacotron uses a Reference Encoder (multiple Convolutional layers and one GRU layer) to extract emotional representations from Mel-Spectrogram, and all modules are trained together. In our work, the architecture of our Emotion Extractor is identical to that of the Reference Encoder. Additionally, for the rest parts, we maintain the original model configuration. Therefore, the only difference is that during the training of the TTS model, we use a pre-trained and frozen Emotion Extractor to provide emotional representations.

The training pipeline is visualized in Fig. 2. We only give a short description of each module here but refer the readers to the original paper [15] for a more in-depth description. The Phoneme Encoder is to process the input phoneme sequence. We use an attention module (Reference Attention) to align the

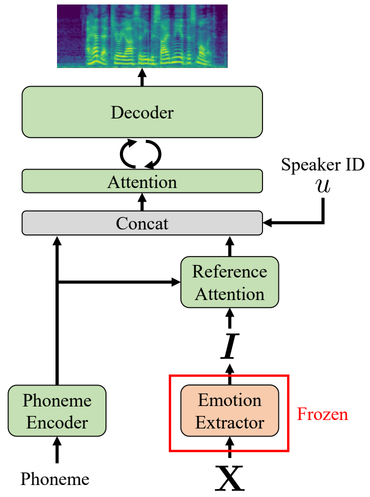

Figure 2: *The training of our TTS model, a pre-trained Emotion Extractor is frozen and combined to provide emotional representations from the input speech* $\mathbf{X}$.

phoneme sequence and the emotional representation $\boldsymbol{I}$, since their lengths are different. Additionally, speaker ID u is mapped into an embedding to represent different speaker characteristics. Afterward, the speaker embedding, phoneme, and emotional representation are concatenated together and sent to the decoder. With another attention module, the decoder aggressively generates the final Mel-Spectrogram.

2.3. Training Details

SER: For the SER task, we train our Emotion Extractor with Adam optimizer for 30k iterations, and set the learning rate to 1e-6.

Emotional TTS: For this task, we first train our Emotion Extractor, then the pre-trained Emotion Extractor is frozen during the training of the TTS model. We train our Emotion Extractor for 30k iterations, and set the learning rate to 1e-6. The TTS model is trained using a 1e-4 learning rate over 600k iterations. Adam optimizer is used for both steps.

3. Experiments

3.1. Experiment Setup

3.1.1. Dataset

SER: EmoV-DB [27], IEMOCAP [28], and CREMA-D [29] are three datasets used for our SER task. EmoV-DB contains 4 speakers and five emotions (Angry, Sleepy, Neutral, Disgusted and Happy), the overall speech samples are around 7 hours. IEMOCAP has 5 speakers, as many other SER works [9, 10], we only focus on four classes (Angry, Sad, Neutral and Happy), resulting in 5531 speech utterances of about 7 hours total duration. CREMA-D contains 6 emotions (Angry, Sad, Neutral, Disgusted, Fear and Happy), amounting to 7742 speech utterances of about 4 hours. Furthermore, for all datasets, in order to simulate the data imbalance issue, we follow [10] and randomly remove 80% of each class except Neutral.

Emotional TTS: For the Emotional TTS task, we employ ESD dataset [30]. ESD has 5 emotions (Neutral, Angry, Sad, Happy and Surprised). In our experiments, we only use the data from 10 English speakers, with each speaker reading 350 utterances for each emotion. To simulate the data imbalance issue, during the training, for each speaker, we only keep 10 utterances (randomly selected) for each emotion class except Neutral.

Data Preprocessing: All speech samples from the aforementioned datasets are resampled to 16kHz. We use the same acoustic features as in [10]. Specifically, to extract Mel-Spectrograms from waveform files, we use a 50-millisecond window and a 50 percent overlap ratio, with 128 Mel Coefficients.

3.1.2. Baselines

SER: The baseline approach we use for the SER task is from [10], where the authors use a GAN-based model to perform automatic augmentation to enhance the SER performance for imbalanced datasets. We follow the same experimental setup and model configurations to ensure a fair comparison.

Emotional TTS: We employ RFTacotron [15] as our TTS baseline model. RFTacotron uses a Reference Encoder to extract a sequence of vectors as the emotional representation from reference speech, this representation is further applied to influence speech generation. In our experiments, we replace the outputs of the Reference Encoder with the representations extracted from our Emotion Extractor. If our representations are more effective, then the synthesized speech should be more expressive.

3.1.3. Evaluation Setup

SER: We conduct 5-fold cross-validation on all SER datasets to fully utilize the data and produce low-biased results. In this work, like many others [9, 10], we report Unweighted Average Recall (UAR) for imbalanced datasets. Precision would be of less interest in our case, since we focus on False Negative (when minority samples are misclassified as a majority), rather than False Positive (when majority samples are misclassified as a minority).

Emotional TTS: We use HiFi-GAN [31] to convert generated Mel-Spectrograms from the TTS models to waveforms. To perform subjective tests, for each emotion, we randomly select 15 unseen text utterances and use another 15 unseen speech samples as references. Thus, 75 speech samples can be synthesized from each model. We conduct subjective evaluations on Amazon Mechanical Turk, and 20 subjects participate in the subjective evaluations. Demo page can be found at `https://anonymous.4open.science/w/INTERSPEECH2023-F8C4/`

3.2. SER Results

3.2.1. Ablation Study

We first perform experiments to investigate the impact of each loss from Eq. 6. The UAR results are listed in Tab. 1. As we can see, latent-level Mixup augmentation performs better than raw-level Mixup (the first two rows). The SER performance improves when we combine both Mixup augmentations, demonstrating that the model profits from both types of mixing and produces more effective and robust emotional representations. After we add $\mathcal{L}_{sim}$ (Eq. 5), the model can exceed all previous models (the last row). Such results demonstrate that forcing the representations obtained from raw- and latent-level Mixup to be similar, improves generalization performance and mitigates the impact of train-test distribution shift, as we claimed before.

According to the result of the Ablation Study, in the rest experiments, we only use models trained with $\mathcal{L}_{total}$.

Table 1: *Ablation Study (UAR Results) on an 80%-reduced imbalanced IEMOCASP dataset.*

Training with	Angry	Sad	Happy	Average
only $\mathcal{L}_{Rmix}$	54.86	55.93	50.06	53.62
only $\mathcal{L}_{Lmix}$	57.12	59.88	53.50	56.83
$\mathcal{L}_{Rmix} + \mathcal{L}_{Lmix}$	59.74	60.09	54.56	58.13
$\mathcal{L}_{total}$	**61.33**	**62.73**	**58.82**	**60.29**

Table 2: *UAR results on three imbalanced datasets. The emotion classes except Neutral of all datasets are reduced by 80%.*

	EmoV-DB		IEMOCAP		CREAM-D	
Emotion	[10]	Ours	[10]	Ours	[10]	Ours
Angry	87.6	**90.5**	**62.8**	61.3	73.9	**74.1**
Sad	-	-	58.1	**62.7**	44.2	**46.9**
Happy	84.6	**86.9**	53.7	**58.8**	39.2	**42.6**
Sleepy	79.8	**83.6**	-	-	-	-
Disgusted	82.0	**84.1**	-	-	44.8	**44.8**
Fear	-	-	-	-	34.3	**41.0**
Average	83.5	**86.3**	58.2	**60.9**	47.3	**49.9**

3.2.2. Comparison with the Baseline

We compare our model with the baseline [10] and list the UAR results in Tab. 2. The left is the results from the EmoV-DB dataset, our model outperforms the baseline for each emotion class, and is around 3% better on average. The results of the IEMOCAP dataset are in the middle. Despite the baseline being better than our model for Angry emotion, our model performs better than the baseline for the other two classes by at least 2%, which yields an average UAR performance that is about 2% higher than the baseline. The UAR results of CREAM-D are listed on the right side. CREAM-D is more challenging than EmoV-DB and IEMOCAP, since it is smaller and contains more emotion classes and speakers. Nevertheless, similar to EmoV-DB and IEMOCAP, our model significantly exceeds the baseline and outputs a performance that is almost 3% greater than the baseline. It is worth mentioning that the Angry class achieves much higher UAR than the other classes. A likely reason for this is that speakers read this emotion louder than other emotions, which makes the Angry class easier to recognize.

From the results of three imbalanced datasets, we observe that the emotional representations extracted by our model are more generalizable and bring benefit for the SER performance on imbalanced datasets.

3.3. Emotional TTS Results

3.3.1. Preference Test

We carry out subjective A/B preference tests to evaluate the models' abilities of emotional expressiveness. Given a pair of synthesized speech samples and one reference, subjects are asked to select the one that conveys emotion more clearly and has a speaking style close to the reference. If there is no detectable difference, they should choose "Same".

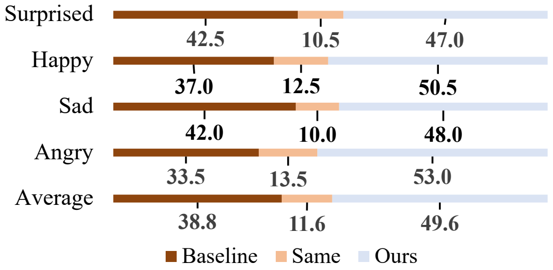

Figure 3: *Preference test for emotion expressiveness. Both TTS models are trained on an imbalanced ESD dataset.*

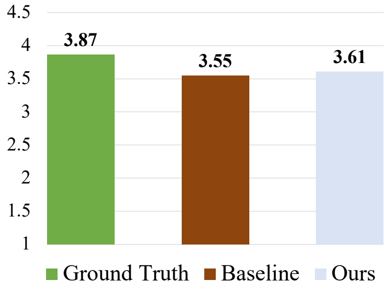

Figure 4: *Naturalness MOS Scores.*

The results are shown in Fig. 3. As we can see, our model outperforms the baseline for all emotions and subjects are barely confused, which suggests that (1) our model's emotional expression is more clear than the baseline's; (2) despite sharing the same architecture with the original Reference Encoder in the baseline, the proposed Emotion Encoder is able to extract more effective and more robust emotional representations from an imbalanced dataset.

3.3.2. Naturalness Test

We further evaluate the naturalness of synthesized speech samples. Subjective measurement Mean Opinion Score (MOS) tests are conducted for this evaluation. During the test, subjects score the speech's naturalness on a scale of 1 (Bad) to 5 (Excellent).

The results are shown in Fig. 4. As we can see, both models are close to the results of the ground-truth speech samples. Moreover, despite using the same architecture, our model slightly outperforms the baseline, indicating that our emotional representations might also bring benefits for the synthesis of natural speech.

4. Conclusion

In this paper, we propose an Emotion Extractor to extract speech emotional representations from imbalanced data. The extracted representations are generalizable and effective. And we apply them to improve the performance of SER and Emotional TTS models. The experimental results show that (1) for the SER task, our model surpasses the state-of-the-art baseline on three imbalanced datasets; (2) the learned representations benefit the TTS model and enable it to generate more expressive speech, even when the training dataset is imbalanced.

5. References

[1] S. Latif, R. K. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. Schuller, "Survey of deep representation learning for speech emotion recognition," *IEEE Transactions on Affective Computing*, 2021.

[2] A. Triantafyllopoulos, B. Schuller, G. .Iymen, M. T. Sezgin, X. He, Z. Yang, P. Tzirakis, S. Liu, S. Mertes, E. Andr'e, R. Fu, and J. Tao, "An overview of affective speech synthesis and conversion in the deep learning era," *ArXiv*, vol. abs/2210.03538, 2022.

[3] J. J. Kim, Y. An, and J. Kim, "Improving speech emotion recognition through focus and calibration attention mechanisms," in *Interspeech*, 2022.

[4] Y. Liu, H. Sun, W. Guan, Y. xin Xia, and Z. Zhao, "Discriminative feature representation based on cascaded attention network with adversarial joint loss for speech emotion recognition," in *Interspeech*, 2022.

[5] Y. Lei, S. Yang, X. Wang, and L. Xie, "Msemotts: Multi-scale emotion transfer, prediction, and control for emotional speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 853–864, 2022.

[6] C.-B. Im, S.-H. Lee, S.-B. Kim, and S.-W. Lee, "Emoq-tts: Emotion intensity quantization for fine-grained controllable emotional text-to-speech," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6317–6321, 2022.

[7] S. Wang, J. Guðnason, and D. Borth, "Fine-grained emotional control of text-to-speech: Learning to rank inter- and intra-class emotion intensities," *ArXiv*, vol. abs/2303.01508, 2023.

[8] A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou, T. Giannakopoulos, A. Katsamanis, A. Potamianos, and S. S. Narayanan, "Data augmentation using gans for speech emotion recognition," in *INTERSPEECH*, 2019.

[9] S. Latif, M. Asim, R. K. Rana, S. Khalifa, R. Jurdak, and B. W. Schuller, "Augmenting generative adversarial networks for speech emotion recognition," *ArXiv*, vol. abs/2005.08447, 2020.

[10] S. Wang, H. Hemati, J. Guðnason, and D. Borth, "Generative data augmentation guided by triplet loss for speech emotion recognition," in *Interspeech*, 2022.

[11] G. Huybrechts, T. Merritt, G. Comini, B. Perz, R. Shah, and J. Lorenzo-Trueba, "Low-resource expressive text-to-speech using data augmentation," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6593–6597, 2020.

[12] M. S. Ribeiro, J. Roth, G. Comini, G. Huybrechts, A. Gabryś, and J. Lorenzo-Trueba, "Cross-speaker style transfer for text-to-speech using data augmentation," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6797–6801, 2022.

[13] M.-J. Hwang, R. Yamamoto, E. Song, and J.-M. Kim, "Tts-by-tts: Tts-driven data augmentation for fast and high-quality speech synthesis," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6598–6602, 2020.

[14] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*, 2018.

[15] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5911–5915, 2018.

[16] H. Zhang, M. Cissé, Y. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *ArXiv*, vol. abs/1710.09412, 2017.

[17] L. Meng, J. Xu, X. Tan, J. Wang, T. Qin, and B. Xu, "Mixspeech: Data augmentation for low-resource automatic speech recognition," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7008–7012, 2021.

[18] X. Zhang, M. Jin, R. Cheng, R. Li, E. Han, and A. Stolcke, "Contrastive-mixup learning for improved speaker verification," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7652–7656, 2022.

[19] W. Kang, J. Alam, and A. Fathan, "L-mix: A latent-level instance mixup regularization for robust self-supervised speaker representation learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1263–1272, 2022.

[20] C. Beckham, S. Honari, A. Lamb, V. Verma, F. Ghadiri, R. D. Hjelm, Y. Bengio, and C. Pal, "On adversarial mixup resynthesis," in *Neural Information Processing Systems*, 2019.

[21] L. Sun, C. Xia, W. Yin, T. Liang, P. S. Yu, and L. He, "Mixup-transformer: Dynamic data augmentation for nlp tasks," *ArXiv*, vol. abs/2010.02394, 2020.

[22] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *ArXiv*, vol. abs/1807.03748, 2018.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *ArXiv*, vol. abs/2002.05709, 2020.

[24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *ArXiv*, vol. abs/2006.09882, 2020.

[25] L. Scheibenreif, J. Hanna, M. Mommert, and D. Borth, "Self-supervised vision transformers for land-cover segmentation and classification," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1421–1430, 2022.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[27] A. Adigwe, N. Tits, K. E. Haddad, S. Ostadabbas, and T. Dutoit, "The emotional voices database: Towards controlling the emotion dimension in voice generation systems," *ArXiv*, vol. abs/1806.09514, 2018.

[28] C. Busso, M. Bulut, C.-C. Lee, E. A. Kazemzadeh, E. M. Provost, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, pp. 335–359, 2008.

[29] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE Transactions on Affective Computing*, vol. 5, pp. 377–390, 2014.

[30] K. Zhou, B. Sisman, R. Liu, and H. Li, "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 920–924, 2020.

[31] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *ArXiv*, vol. abs/2010.05646, 2020.