Interspeech 2024
1-5 September 2024, Kos, Greece

Dataset-Distillation Generative Model for Speech Emotion Recognition

Fabian Ritter-Gutierrez[1,2], *Kuan-Po Huang*[3,4], *Jeremy H.M Wong*[2], *Dianwen Ng*[1,5], *Hung-yi Lee*[3], *Nancy F. Chen*[1,2], *Eng-Siong Chng*[1]

[1]Nanyang Technological University, Singapore [2]Institute for Infocomm Research (I2R), Singapore [3]National Taiwan University, Taiwan [4]ASUS Intelligent Cloud Services, Taiwan [5]Speech Lab of DAMO Academy, Alibaba Group, Singapore

s220064@e.ntu.edu.sg, stufarg@i2r.a-star.edu.sg

Abstract

Deep learning models for speech rely on large datasets, presenting computational challenges. Yet, performance hinges on training data size. Dataset Distillation (DD) aims to learn a smaller dataset without much performance degradation when training with it. DD has been investigated in computer vision but not yet in speech. This paper presents the first approach for DD to speech targeting Speech Emotion Recognition on IEMOCAP. We employ Generative Adversarial Networks (GANs) not to mimic real data but to distil key discriminative information of IEMOCAP that is useful for downstream training. The GAN then replaces the original dataset and can sample custom synthetic dataset sizes. It performs comparably when following the original class imbalance but improves performance by 0.3% absolute UAR with balanced classes. It also reduces dataset storage and accelerates downstream training by 95% in both cases and reduces speaker information which could help for a privacy application.

Index Terms: self-supervised learning, dataset distillation, speech emotion recognition, generative adversarial network.

1. Introduction

End-to-end (E2E) machine learning and self-supervised learning (SSL) techniques have revolutionized speech processing in various tasks [1–3]. However, they rely on large data resources for training, posing storage and data processing challenges. For example, [4] utilized 180k hours of labelled data and required 20 days of training on 64 GPUs to train a single model. Such data-intensive models present financial and logistical challenges when faced with limited resources while posing severe environmental impact [5]. Despite these issues, the current training paradigm necessitates a vast amount of data [6].

Dataset Distillation (DD) [7] has emerged, showing great promise for reducing training costs. DD aims to learn discriminative and informative samples and form a smaller synthetic dataset hoping to keep as much performance as the original one. DD deviates from the "data-selection" paradigm [8] where a smaller dataset is created by selecting representative data points in the dataset. In contrast, DD learns abstract representations that convey the dataset's most discriminative information, which may or may not look realistic.

DD is a popular emerging paradigm in Computer Vision (CV) [9–13] yet it has not been explored for speech processing tasks. DD for speech processing introduces unique challenges due to the inherent differences between speech signals and images. Speech is a temporal signal with temporal dependencies. Hence, there is relevant information to distil across time. This paper proposes a first attempt of DD on speech processing task, aiming to 1) significantly reduce the disk storage requirement compared to the original dataset, 2) reduce training time computation on the downstream task, 3) make speaker identity harder to recover to enhance privacy and 4) alleviate data label imbalance. Such goals should be achieved without considerably hurting downstream model performance when training with the distilled dataset.

Speech emotion recognition (SER) task in the IEMOCAP dataset [14] is chosen as a case study for the following reasons. First, SER is an utterance-level classification task, where the variable length speech sequence is mapped into a single vector for classification. This is a favorable starting point to analyze the feasibility of this research direction on speech processing before extending the approach to speech tasks that make predictions over frames of the speech sequence. Second, while utterance-level classification makes the task more manageable, the subtleties needed to model emotions are challenging and interesting. The DD algorithm will need to convey discriminative information of a speech signal for ER classification.

Fig. 1 shows the usage scenario of the proposed method. Rather than training a downstream model with the original dataset, which requires expensive model training due to hyperparameter tuning, downstream architecture selection, and so on, we propose to learn a distribution that summarizes the training data, and that is controlled only by the emotion class labels. By learning a distribution that summarizes the training data across emotion labels, we do not need to retain a record of the original speech. Hence, our proposal implicitly enhances privacy. Nonetheless, this does not means the proposal guarantees private generated representations. Once this summary distribution is learned as a generative model, a custom budget of samples per class can be generated to train downstream models, perform parameter tuning and so on. While training a generator incurs a cost, our proposal aims to provide a generator that replaces the dataset, meaning that training the generator is a once-for-all process.

The method, depicted in Fig 2, employs a Generative Adversarial Network (GAN) for DD in IEMOCAP, favored over a Diffusion Probabilistic Model (DPM) due to its smaller size, higher computational efficiency, and quicker on-the-fly data generation capabilities [15]. Nonetheless, GANs have been designed to generate real-looking data, differing from our goal of learning a summary distribution of the dataset useful for downstream training. Hence, to make the GAN learn discriminative information useful for downstream performance, we propose to bias the GAN by adding a term that minimizes the Kullback-Leibler (KL) divergence between the softmax probabilities of emotion classes of downstream forward passes between the real and synthetic data. We prevent the GAN from merely memorizing the softmax probability distribution by sampling from a variety of downstream model checkpoints, thereby introducing

10.21437/Interspeech.2024-1430

a range of possible KL divergence targets. Furthermore, a diversity penalty term is added to make the GAN sample more diverse data on smaller synthetic dataset sizes. To test the efficiency of the proposed method, we do ablations to see real data test set performance on IEMOCAP. The results obtained show that our proposal consistently maintains close accuracy performance comparable to a model trained on the real IEMOCAP dataset and it is consistently better than a GAN [16] trained without our proposed criteria with statistical significance at a p-value of 0.05. The proposed method reduces the dataset size and training time by 95% with minimal performance degradation. Additionally, it improves SER over the real data training when our method samples balanced datasets. Hence, the proposal alleviates data imbalance issues inherent in IEMOCAP. Finally, this proposal implicitly decreases speaker identity information which fosters possibilities for privacy related applications.

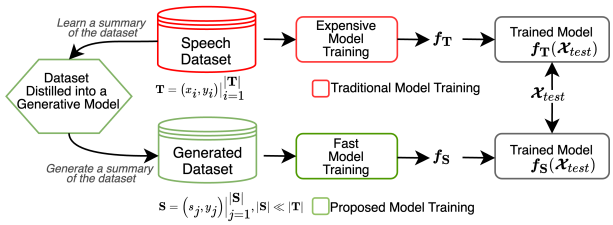

Figure 1: *Usage scenario for DD on speech processing tasks.* $f_{\mathbf{T}}(x_{test})$ *represents inference on a downstream model* f *trained under dataset* $\mathbf{T}$.

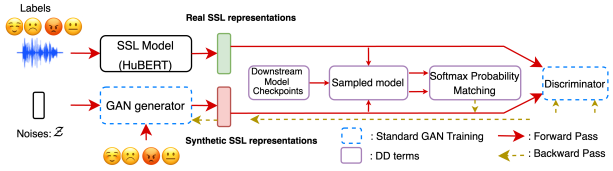

Figure 2: *Schematic representation of the proposed DD. The blue dashed lines represent the standard training of a GAN*

2. Related Work

While there is no work directly aiming to distill a dataset for SER or any speech processing task, there is work that leverages GANs for data augmentation in SER. In [17], an unconditional and conditional GAN was trained for the IEMOCAP dataset. [18], uses a conditional GAN to do mel-spectrogram augmentation to improve performance on less representative emotion classes for IEMOCAP. [19] investigates CycleGAN for emotion style transfer, aiming to generate realistic emotion data. The study adds an evaluation of real test sets for models trained on synthetic data only, revealing a performance gap above 8% between training on real versus synthetic data. There are more similar works using GANs for data augmentation such as [20–22] but with different GANs architectures and some recent work has attempted speech emotion recognition data augmentation using denoising diffusion probabilistic models (DDPM) [23].

3. Dataset Distillation

Generally speaking, DD aims to learn a small dataset that achieves comparable performance to the original dataset that it is distilling from. Let $\mathbf{T} = (\mathbf{x_i}, \mathbf{y_i})|_{i=1}^{|\mathbf{T}|}$ be the real dataset consisting on data-label pairs $\mathbf{x}_i, \mathbf{y}_i$ with $\mathbf{x}_i \in \mathbb{R}^d$ with d the feature dimension and $\mathbf{y}_i \in \mathbb{R}^c$ with c the number of classes. DD aims to create a synthetic dataset $\mathbf{S} = (\mathbf{s}_j, \mathbf{y}_j)|_{j=1}^{|\mathbf{S}|}$ ($|\mathbf{S}| \ll |\mathbf{T}|$). Once $\mathbf{S}$ is learned, the dataset is deployed to train a downstream model, and that model is evaluated on the real data test set.

DD methods are based on three strategies [7]: i) performance matching: monitoring performance achieved by a neural network with the original dataset versus the synthetic dataset [9]; ii) gradient matching: match the gradient in a neural network of the original and synthetic dataset at each iteration [10]; iii) distribution or feature matching: match the features produced on a neural network for the real and synthetic data [11, 12]. In general, the algorithm will define a fixed budget of number of elements per class when doing DD. Hence, if a different budget is needed, a whole DD training must be done again. Differently, the works [24, 25] distill CV datasets into a generative model. Hence, rather than directly learning a dataset $\mathbf{S}$, they learn a generator $\boldsymbol{g}$ that can sample different datasets based on a sample per class budget. Our proposal is motivated by these ideas.

4. Dataset Distillation for Speech Emotion

In this paper, rather than directly learning a dataset $\mathbf{S}$, a generative model $\boldsymbol{g}$ is learnt to generate summary distributions of $\mathbf{T}$. Once $\boldsymbol{g}$ is learnt, custom-defined samples per emotion can be generated. Small-size generative models are designed, thereby significantly decreasing storage requirements of the original dataset as seen in Table 1.

The proposed approach consists of two stages, the first stage is a standard GAN training, particularly the conditional Wasserstein GAN implementation with gradient penalty (WGAN-GP) [16]. In WGAN-GP, the discriminator $\boldsymbol{d}_\omega$, with ω the weight parameters, is optimized as,

$$\mathrm{L}_{\mathrm{D_{ADV}}}(w) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim P(\boldsymbol{z})}\left[\boldsymbol{d}_\omega\left(\boldsymbol{g}_\phi(\boldsymbol{z})\right)\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim P(\boldsymbol{x})}\left[\boldsymbol{d}_\omega(\boldsymbol{x})\right] + \lambda_1 \mathrm{L}_{\mathrm{GP}}(w), \quad (1)$$

where $\boldsymbol{g}_\phi$ is the generator parametrized by the weights ϕ, $P(\boldsymbol{z})$, $P(\boldsymbol{x})$ denotes the distribution of noise (latent) vectors and real samples respectively. A noise vector $\mathbf{z} \sim P(\boldsymbol{z})$ contains the information of the label $\mathbf{y}$ in the form of a one-hot vector, i.e. $\mathbf{z} \equiv [\mathbf{y} \oplus \mathbf{e}]$, with $\oplus$ the concatenation operation and $\mathbf{e} \sim \mathcal{N}(0, 1)$.

The gradient penalty $\mathrm{L}_{\mathrm{GP}}(w)$ is needed to have a valid Wasserstein distance computation and λ_1 controls the importance of this term. We use $\lambda_1 = 10$ as in the original WGAN-GP [16].

The generator in WGAN-GP is trained to minimize,

$$\mathrm{L}_{\mathrm{G_{ADV}}}(\phi) = -\mathop{\mathbb{E}}_{\boldsymbol{z}\sim P(\boldsymbol{z})}\left[\boldsymbol{d}_w\left(\boldsymbol{g}_\phi(\boldsymbol{z})\right)\right]. \quad (2)$$

Additionally, motivated by speech processing research on mel-spectrogram inversion [26, 27], we add a feature matching (FM) loss, shown to improve stability for generator training. The FM loss is defined as,

$$\mathrm{L}_{\mathrm{FM}}\left(\boldsymbol{g}_\phi, \boldsymbol{d}_w\right) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim P(\boldsymbol{x}) \\ \boldsymbol{z}\sim P(\boldsymbol{z})}}\left[\sum_{l=1}^{M} \frac{1}{M}\left|\boldsymbol{d}_w^{(l)}(\boldsymbol{x}) - \boldsymbol{d}_w^{(l)}(\boldsymbol{g}_\phi(\boldsymbol{z}))\right|\right], \quad (3)$$

where $\boldsymbol{d}_w^{(l)}$ is the feature map at the "l-th" layer of the discriminator $\boldsymbol{d}_w$, and M the number of layers. Eq. (3) helps the generator to sample features on the same space than the real data.

Finally, to use the conditioning class label information, a cross-entropy loss is added. Then, the final loss for the discriminator is,

$$\mathrm{L_D} = \mathrm{L_{D_{ADV}}}(\omega) + \lambda_2 \mathop{\mathbb{E}}_{\boldsymbol{x}\sim P(\boldsymbol{x})} \Big[\mathrm{CE}(\boldsymbol{d}_w^{\text{class}}(\boldsymbol{x}), \boldsymbol{y})\Big] + \lambda_3 \mathop{\mathbb{E}}_{\boldsymbol{z}\sim P(\boldsymbol{z})} \Big[\mathrm{CE}(\boldsymbol{d}_w^{\text{class}}(\boldsymbol{g}_\phi(\boldsymbol{z})), \boldsymbol{y})\Big], \quad (4)$$

with CE denoting the cross-entropy loss, $\boldsymbol{d}_w^{\text{class}}(\cdot)$ the logits distribution of the emotion classes $\boldsymbol{y}$, and λ_i represent scalar weights. The final generator loss is,

$$\mathrm{L_G} = \mathrm{L_{G_{ADV}}}(\phi) + \lambda_3 \mathop{\mathbb{E}}_{\boldsymbol{z}\sim P(\boldsymbol{z})} \big[\mathrm{CE}(\boldsymbol{d}_w^{\text{class}}(\boldsymbol{g}_\phi(\boldsymbol{z})), \boldsymbol{y})\big] + \lambda_4 \mathrm{L_{FM}}. \quad (5)$$

Eq. (4) and (5) are designed to generate data that resembles real instances as done in previous work [18, 19]. The aim of this paper diverges from conventional uses of GANs for creating real-looking data. Instead, the focus is on harnessing GANs to generate key discriminative information that serves downstream model performance so that it can be used for DD. This point is important as it differs from the paradigm of generating the same distribution of the original dataset but rather a distribution that contains the information useful for downstream task training. To achieve this goal, a softmax probability matching method is proposed to minimize the KL-divergence between the softmax probabilities of real and synthetic data across a range of downstream model checkpoints, this range is needed to avoid the GAN memorizing the logits distribution of a single model. The proposed softmax matching loss (SML) enforces the generator $\boldsymbol{g}_\phi$ to generate representations that are useful for downstream model training. Specifically, let $\boldsymbol{\Theta}$ consist of a distribution of model checkpoints. For any sampled model $\boldsymbol{f}_\theta$ from this set, where $\theta \sim \boldsymbol{\Theta}$ represents the downstream model weights, the SML is defined as,

$$L_{\text{SML}} = \frac{1}{B} \sum_{j=1}^{B} \sum_{i=1}^{|\boldsymbol{y}|} \boldsymbol{f}_\theta(\mathbf{x_j})_i \log \left(\frac{\boldsymbol{f}_\theta(\mathbf{x_j})_i}{\boldsymbol{f}_\theta(\boldsymbol{g}_\phi(\boldsymbol{z}_j))_i} \right), \quad (6)$$

for $|\boldsymbol{y}|$, the number of classes, B the batch size and $\boldsymbol{f}_\theta(\cdot)_i$ is the softmax probability of the i-th class given some real observation $\mathbf{x_j}$ or generated representation $\boldsymbol{g}_\phi(\boldsymbol{z}_j)$.

Furthermore, inspired by [28], a diversity penalty is included into the generator $\boldsymbol{g}_\phi$ to encourage the generation of a wider variety of samples. Rather than producing samples clustered around a mode, the goal is to span the support of the real data as broadly as possible. The diversity penalty loss is defined as,

$$\mathrm{L_{DIV}}(\boldsymbol{g}_\phi) = - \mathop{\mathbb{E}}_{\boldsymbol{z}1,\boldsymbol{z}2\sim P(\boldsymbol{z})} \left[\min\left(\frac{|\boldsymbol{g}_\phi(\boldsymbol{z}1) - \boldsymbol{g}_\phi(\boldsymbol{z}2)|}{|\boldsymbol{z}1 - \boldsymbol{z}2|}, \tau\right)\right], \quad (7)$$

with τ a scalar that bounds the diversity penalty for stability. Eq. (7) compares noises of the same class. Then, for two vectors $\mathbf{z}_1$ and $\mathbf{z}_2$, if $\mathbf{z}_1 \approx \mathbf{z}_2$, the generator should generate two similar vectors. On the other hand, if the noises are different, then the generator should generate a different representation, thus avoiding mode collapse.

Finally, the proposed DD method consists of the same discriminator loss of Equation (4) and the following generator loss,

$$L_{\mathrm{G_{DD}}} = L_{\mathrm{G}} + \lambda_5 \mathrm{L_{DIV}} + \lambda_6 L_{\text{SML}}. \quad (8)$$

Table 1: *DD size reduction of training set of IEMOCAP*

	Audio Files	SSL feats	GAN Size	Size Reduction (%) Audio	SSL feats
GAN-CNN	1.8 GB	2.4 GB	0.1 GB	94.44	95.83
GAN-ATT			0.06 GB	96.95	97.50

5. Experiments

5.1. Implementation details

Dataset and SSL setup: As explained in Section 1, SER task is chosen and SUPERB [29] framework is followed for easy reproducibility and comparison with real data training. Experiments follow the leave-one-out session and only leave-out Session 1 is assessed due to computational resource restrictions. Nonetheless, in order to account for the possible variance in the results, McNemar's test is conducted at a p-value of 0.05 to verify statistical significance. The training data consists of Session 2 to 4, spanning 3,556 audios to distill. Motivated by [17,19,30] that does GAN data augmentation on a time averaged openSMILE [31] representation, this work generates a distribution on SSL representations but retaining the time dimension. Distillation is done over HuBERT Base [1] SSL representations and evaluations are done with Unweighted Average Recall (UAR) to account for class imbalance.

Discriminator and Generator Architecture: Two small size architectures are considered. The first, named GAN-CNN, is a WGAN-GP model utilizing solely convolutional layers (CNN) for both its discriminator and generator. The discriminator is composed of 8 2D-CNN layers, each featuring layer normalization and leaky-relu activation. The final CNN layer connects to two feed-forward layers: one calculates the Wasserstein distance, and the other predicts the class category. The generator in GAN-CNN employs 2D CNN and transposed convolution layers followed by batch normalization and leaky-relu activation. There is no tanh operation at the generator's output, because the original SSL features are not limited to the [-1,1] range.

Using only CNN layers for the generator has the inductive bias that points that are spatially close to each other are correlated while neglecting long-range correlations. Nonetheless, for SER, modeling a long temporal context over each feature dimension may be important. Hence, the generator is modified to include only one self-attention operation over the time dimension after the 4-th CNN layer. To reduce number of parameters even further, the number of channels in the CNN layers are reduced from 256 to 128 and dilation is included to increase the receptive field of each CNN layer. This architecture is called GAN-ATT. Both models train on an A100-SXM4-40GB GPU, requiring 30 GPU hours each. Table 1, shows the sizes of the two GAN's architectures, highlighting a nearly 95% size reduction compared to the original IEMOCAP audio. Such results in Table 1 are important when scaling up to bigger datasets.

5.2. GAN as a dataset distillator

Table 2 analyzes the effect of training with a traditional WGAN-GP (Baseline) versus the proposed losses $\mathrm{L_{DIV}}$ and $\mathrm{L_{SML}}$ for the two generator architectures mentioned. DD aims to learn key discriminative information for training. In order to evaluate the efficiency of the discriminative information modelled, it is common to analyze DD performance using a small number of datapoints. Therefore, Table 2 analyzes such results under a low points per class (ppc) budget of 50 ppc (5.6% of the size of the original dataset) and 100 ppc (11.2% of the size of the original dataset). Additionaly, to analyze how the proposal scales to

Table 2: *SER UAR (%↑) for downstream model trained only with generated data. Two generators are evaluated, GAN-CNN and GAN-ATT, under 50 points per class (ppc) and 100 ppc. Baseline denotes the GAN without DD criterions. † denotes a McNemar's test statistically significant difference over the Baseline.‡ denotes significance over the $+L_{SML}$ model.*

Model	GAN-CNN		GAN-ATT			
ppc	50 UAR	100 UAR	50 UAR	100 UAR	800 UAR	1800 UAR
Baseline	47.75	53.01	56.31	59.07	61.44	62.05
+ L_{DIV}	48.52	53.25	58.89	60.52	62.20	62.86
+ L_{SML}	53.79†	60.52†	56.99†	60.26†	63.96†	64.07†
+ L_{DIV} + L_{SML}	**54.99**†‡	**60.99**†‡	**60.27**†‡	**61.95**†‡	**64.35**†‡	**64.70**†‡

Table 3: *UAR (%↑) test performance for real data training (Real SSL) and for GAN-ATT trained for balanced and imbalanced class labels distribution scenarios.*

Method	2447 points Balanced	2447 points Imbalanced	Full data Imbalanced	Full data Balanced
Real SSL	64.09	63.59	64.20	-
GAN-ATT	64.47	63.21	63.69	**64.50**

bigger data samples, results with 800 ppc and 1800 ppc are included for GAN-ATT. All results are evaluated on the real data of Session 1 in IEMOCAP. From Table 2, it is evident that incorporating the proposed L_{DIV} and L_{SML} into the Baseline WGAN-GP significantly improves UAR across both generator architectures (GAN-CNN and GAN-ATT) and for both 50 ppc and 100 ppc dataset budgets. Furthermore, it can be seen that both terms are complementary. When comparing the Baseline with models using either L_{DIV} or L_{SML} individually, there is a noticeable improvement in performance. For instance, in the GAN-CNN architecture at 50 ppc, the UAR improves from 47.75% to 48.52% with L_{DIV} and to 53.79% with L_{SML} and to 54.99% when both terms are used together. For the GAN-ATT architecture, the trends are analogous. Interestingly, the GAN-ATT generator using both terms proposed for a budget of 11.2% of the original dataset size can achieve an UAR score of 61.95% which is only 2.25% less than the model trained with the full original training data (see first row in Table 3). For bigger data budgets, GAN-ATT surpasses the performance of the model trained with the original training set. Notably, two-tailed McNemar's test is performed at a p-value of 0.05 and results shows statistical significance for the L_{SML} and $L_{DIV} + L_{SML}$ models when compared to the Baseline. Additionally, the Baseline $+ L_{DIV} + L_{SML}$ model is also statistically significant when compared with $+L_{SML}$ only.

Table 3 compares GAN-ATT's performance against real data training in both balanced and imbalanced scenarios. IEMOCAP is a well known imbalanced dataset, meaning that some classes are represented more than others. Training with imbalanced data may hurt performance and hence using a GAN to alleviate this issue may be of importance. Last column of Table 3 shows that using the same size than the original dataset but with balanced classes improves performance than training with the original dataset. On the other hand, Full data Imbalanced column assess GAN-ATT under the same class label distribution of the original train set which shows similar performance than the model trained with real SSL. Besides, in order to test the real data set in a balance scenario, we select all the datapoints from the minority emotion class (693 utterances) and randomly select 693 utterances for each of the rest of emotion classes. Similarly, we analyze performance of 693 ppc for the GAN-ATT (2447 points in total) and finally we see performances of real SSL and GAN-ATT under the imbalance scenario for 2447 datapoints. Findings in Table 3 suggest that the proposed method can be used to alleviate data label imbalance because GAN only training can improve performance versus real data training. Such results suggest that having a generative model that can modify the train data class label distribution is beneficial and is a strength of this proposed method. Finally, we noticed that using GAN data makes the downstream model quickly converge on the real validation set, making the model to be trained in less than 5 minutes. On the other hand, real data training convergence is slower, taking around 90 minutes to train which is nearly a 95% time reduction for downstream model training. This efficiency facilitates quicker hyperparameter optimization for downstream models, showcasing another advantage of our approach.

Table 4: *SID Accuracy (Acc) with different speaker embeddings.*

Speaker Embedding	Acc (%↓)
SID downstream model	80.89
SER downstream model with Real SSL	44.87
SER downstream with Baseline WGAN-GP	42.87
SER downstream with Proposed GAN-ATT	**37.99**

5.3. On the privacy aspect

Although this method does not inherently guarantee privacy, its use of GANs to learn SSL-like representations, conditioned solely on emotion labels, does not seem optimal to retain other forms of information. This section focuses on speaker identity, but similar arguments can be made about the retention of information such as content. The model's design, results in the generation of abstract representations that enhance downstream model performance for SER. This implicitly bolsters privacy by limiting the frame-level information necessary for accurate speech reconstruction. To assess the potential for retaining speaker information, we propose testing using the downstream model's first layer as a speaker embedding, a technique widely recognized in speaker identification (SID) studies [32, 33]. Table 4 shows such results for SUPERB SID task, where our proposed method reduces speaker information by 6.88% compared to the linear layer of a downstream model trained for SER with real SSL representations. While these results do not mean the GAN-ATT ensures privacy, it does mean there is an implicit reduction on speaker identity modelling which could serve as a starting point for explicitly training DD that ensures privacy. This will be investigated in future work.

6. Conclusions

This study introduced DD for SER by leveraging a GAN to generate datasets that are useful for downstream model training. A softmax probability matching loss is proposed to achieve such goal. Diversity penalty is proposed to sample more variety of synthetic datapoints. The method achieves performance on par compared to real data downstream model training while substantially reducing dataset size and downstream training time. Our method can alleviate data label imbalance. Our method as well carries less speaker information which could serve as a starting point for an application on privacy preserving dataset distillation. Future work will analyze this direction as well as scaling to bigger datasets.

7. Acknowledgments

I want to deeply thank my friend Nikita Kuzmin for the great discussions on the privacy aspect. Unfortunately, I end up not adding such results on this manuscript.
The computational work for this article was fully performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

8. References

[1] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," in *IEEE/ACM TASLP*, 2021.

[2] S. Chen, C. Wang, Z. Chen, Y. Wu, *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," in *IEEE J-STSP*, 2021.

[3] A. Mohamed, H. yi Lee, L. Borgholt, J. D. Havtorn *et al.*, "Self-supervised speech representation learning: A review," in *IEEE J-STSP*, 2022.

[4] Y. Peng, J. Tian, B. Yan *et al.*, "Reproducing whisper-style training using an open-source toolkit and publicly available data," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[5] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *ACL*, 2019.

[6] J. Droppo and O. H. Elibol, "Scaling laws for acoustic models," in *Interspeech*, 2021.

[7] R. Yu, S. Liu, and X. Wang, "Dataset distillation: A comprehensive review," *IEEE TPAMI*, 2023.

[8] K. Killamsetty, X. Zhao, F. Chen, and R. Iyer, "Retrieve: Coreset selection for efficient and robust semi-supervised learning," *NeurIPS*, 2021.

[9] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv preprint*, 2018.

[10] B. Zhao, K. R. Mopuri, and H. Bilen, "Dataset condensation with gradient matching," *ICLR*, 2021.

[11] B. Zhao and H. Bilen, "Dataset condensation with distribution matching," in *WACV*, 2023.

[12] K. Wang, B. Zhao, X. Peng, Z. Zhu *et al.*, "Cafe: Learning to condense dataset by aligning features," in *CVPR*, 2022.

[13] D. Zhou, K. Wang, J. Gu, X. Peng, D. Lian *et al.*, "Dataset quantization," in *ICCV*, 2023.

[14] C. Busso, M. Bulut, C.-C. Lee, E. A. Kazemzadeh *et al.*, "Iemocap: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, 2008.

[15] X. Zhang, J. Wang, N. Cheng, and J. Xiao, "Voice conversion with denoising diffusion probabilistic gan models," in *ADMA*, 2023.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *NeurIPS*, 2017.

[17] S. Sahu, R. Gupta, and C. Espy-Wilson, "On enhancing speech emotion recognition using generative adversarial networks," in *Interspeech*, 2018.

[18] A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou *et al.*, "Data augmentation using gans for speech emotion recognition." in *Interspeech*, 2019.

[19] B. Fang, N. Michael, and V. N. Thang, "Cyclegan-based emotion style transfer as data augmentation for speech emotion recognition." in *Interspeech*, 2019.

[20] Y. Lu and M. Man-wai, "Adversarial data augmentation network for speech emotion recognition," in *APSIPA ASC*, 2019.

[21] L. Yi and M. wai Mak, "Improving speech emotion recognition with adversarial data augmentation network," in *IEEE TNNLS*, 2022.

[22] H. Xiangheng, C. Junjie, R. Georgios, and S. B. W, "An improved stargan for emotional voice conversion: Enhancing voice quality and data augmentation," in *Interspeech*, 2021.

[23] I. Malik, S. Latif, R. Jurdak, and B. Schuller, "A preliminary study on augmenting speech emotion recognition using a diffusion model," in *Interspeech*, 2023.

[24] B. Zhao and H. Bilen, "Synthesizing informative training samples with gan," in *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

[25] K. Wang, J. Gu, D. Zhou, Z. H. Zhu, W. Jiang, and Y. You, "Dim: Distilling dataset into generative model," *ArXiv*, 2023.

[26] K. Kumar, R. Kumar, T. de Boissière *et al.*, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *NeurIPS*, 2019.

[27] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *NeurIPS*, 2020.

[28] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," in *ICLR*, 2019.

[29] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. Lai, K. Lakhotia *et al.*, "Superb: Speech processing universal performance benchmark," in *Interspeech*, 2021.

[30] J. Han, Z. Zhang, Z. Ren, F. Ringeval, and B. Schuller, "Towards conditional adversarial training for predicting emotions from speech," *ICASSP*, 2018.

[31] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *ACM Multimedia*, 2013.

[32] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP*, 2018.

[33] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Disentangling voice and content with self-supervision for speaker recognition," *NeurIPS*, 2024.