Self-FiLM: Conditioning GANs with self-supervised representations for bandwidth extension based speaker recognition

Saurabh Kataria[1,2], *Jesús Villalba*[1,2], *Laureano Moro-Velázquez*[1],
Thomas Thebaud[1], *Najim Dehak*[1,2]

[1]Center for Language and Speech Processing, [2]Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD, USA

{skatari1,jvillal7,laureano,tthebau1,ndehak3}@jhu.edu

Abstract

Speech super-resolution/Bandwidth Extension (BWE) can improve downstream tasks like Automatic Speaker Verification (ASV). We introduce a simple novel technique called Self-FiLM to inject self-supervision information in existing BWE models via Feature-wise Linear Modulation. We hypothesize that such information contains domain/environment information, which can help BWE deliver zero-shot generalization. Self-FiLM improves conditional GAN-based BWE by 18% (relative) in Equal Error Rate and 8.5% in minimum Decision Cost Function on the x-vector & Probabilistic Linear Discriminant Analysis based state-of-the-art ASV system on SRE21 test. We further improve it by using deep feature losses from time-domain models and re-training data2vec 2.0 models on naturalistic wideband (VoxCeleb) and telephone data (SRE Superset etc.). Lastly, we integrate Self-FiLM in CycleGAN to obtain a completely unsupervised solution that matches the CGAN-based semi-supervised performance.

Index Terms: Self-supervision, FiLM conditioning, conditional GAN, super-resolution, CycleGAN, data2vec 2.0

1. Introduction

Deep learning has incredibly advanced speech applications like source separation, speech enhancement, and Bandwidth Extension (BWE) [1]. Such inverse problems involve representation learning and feature mapping between domains. The rise of Self-Supervised Learning (SSL) has called for a joint investigation of SSL and BWE (our application of interest). Conditional variants of Generative Adversarial Network (GAN) [1], and diffusion model have shown great promise [2] for BWE. We aim to bridge the gap between SSL and deep generative models by learning to condition GAN with SSL representations.

SoundFilter [3] is a one-shot source separation model which uses a short target utterance as conditioning using Feature-wise Linear Modulation (FiLM) [4]. In [5], authors use a pre-trained WavLM [6] SSL model (with additional fine-tuning step) as another input to speech enhancement network. TUNet [7] uses temporal FiLM-based UNet architecture for BWE and simple self-supervision losses but does not use any conditioning information. [8] pursues mixed-bandwidth Automatic Speech Recognition (ASR) by doing channel-aware pre-training in a HuBERT [9]-inspired SSL model.

We develop SSL-conditioned BWE models to assist telephony ASV (downstream task) [10], which prior work still needs to address. Considering SSL representation as a proxy for Acoustic Environment Embedding (AEE), we explore the zero-shot adaptation capability of our system during inference using AEE information. We choose to condition the hidden layers of the BWE model and not provide SSL embedding sequence as an additional input (e.g., to the first layer) to avoid re-designing the BWE-ASV pipeline. Also, we require continuous target prediction for BWE, and thus we avoid directly mapping SSL embeddings to desired temporal output. With our proposed scheme Self-FiLM, we first establish the utility of various pre-trained SSL models with CGAN. We also visualize the conditioning information by speaker, language, and domain label as explored similarly in [11]. Building on an efficient SSL model such as data2vec 2.0, we explore in-domain training as done in Robust wav2vec 2.0 [12]. We also study Self-FiLM with Deep feature Loss (DFL) for speaker preservation and CycleGAN for unsupervised learning, which is unavailable in prior work.

As our main contributions, we 1) propose a simple conditioning technique to utilize self-supervised representations in bandwidth extension to provide test acoustic environment information, 2) visualize the proposed conditioning information by different domain labels, 3) study the effect of using in-domain SSL models by re-training data2vec 2.0 on naturalistic mixed-bandwidth data, 4) demonstrate the compatibility of Self-FiLM with CycleGAN, and deep feature losses.

2. Background

2.1. Self-supervised learning models

We extract 256-D frame-level representations from primarily pre-trained small/BASE versions of 16KHz SSL audio models. since they capture low as well as high-level information [11].

wav2vec 2.0 [13]: This model is trained with a contrastive loss defined over jointly learned quantization of latent representations. The feature encoder consists of seven convolutional blocks in the BASE version (95M parameters). The context/transformer network has 12 layers, 768 model dimensions, 3072 Point-wise Feed-Forward Network (FFN) inner dimension, eight attention heads, and relative positional encoding. The training data is Librispeech [14] which consists of 960h of read speech. To test multi-lingual generalization on our test sets, we also experiment with *Robust Large wav2vec 2.0* (0.3B parameters), which is trained on Libri-light, Common-Voice, Switchboard, and Fisher [12].

XLSR-53 [15]: This is another multi-lingual counterpart of wav2vec 2.0 (BASE) trained with 53 languages from CommonVoice [16] (read speech), BABEL [17] (conversational telephone speech), and Multi-Lingual Librispeech (MLS) [18] (read speech from audiobooks).

WavLM [6]: This model jointly accomplishes masked speech target prediction (like HuBERT [9]) and denoising. The denoising capability makes it more conducive to non-ASR tasks. The BASE model (94.7M parameters) is trained on Librispeech and has similar architecture to wav2vec 2.0.

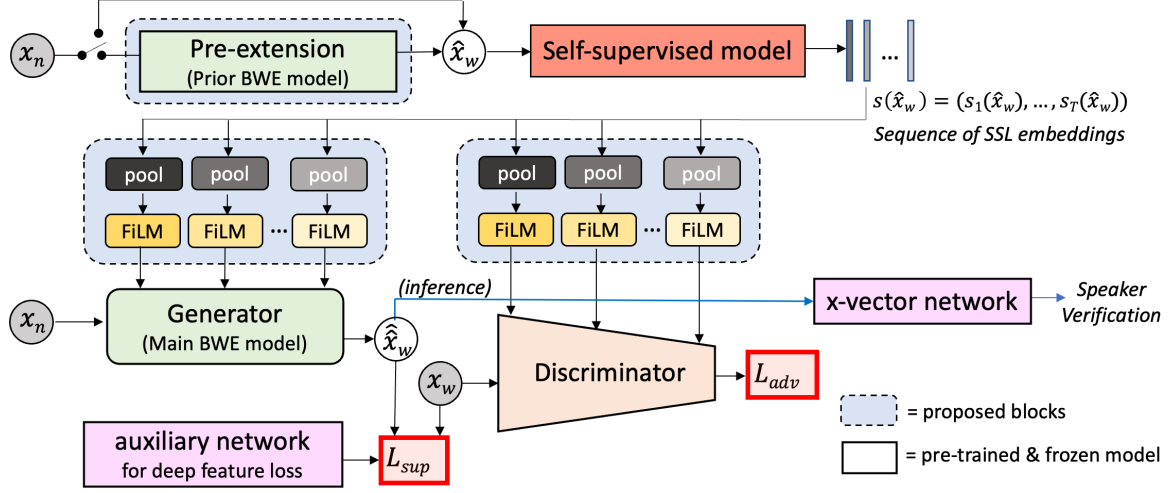**Data2vec 2.0** [19]: This BASE model (93.8M parameters) is

Figure 1: *Illustration of the Self-FiLM framework for Conditional GAN bandwidth extension. The generator and discriminator receive information from the self-supervised model after pooling and FiLM operation. An optional* pre-extension *block is present before the SSL model. An auxiliary network is used for deriving speaker-preserving (deep feature) loss.*

derived from a non-contrastive student-teacher self-distillation wav2vec 2.0 formulation, where the teacher follows a running exponential average of the student in the learning process.

2.2. Embedding pooling methods

mean, **mean+std**: Here, *mean* refers to simple average pooling across time dimension, while *mean+std* (*statistics pooling*) refers to the concatenation of mean and standard deviation.
LDE [20]: In Learnable Dictionary Encoding (LDE), we representations $(\mathbf{s_1}, \dots, \mathbf{s_T})$ are assumed to be in C clusters. We learn a dictionary with one center per cluster. Details for soft cluster assignment and final embedding are in [21].
ScaleAtt: We use a modified form of scaled dot-product attention [22], which we term as ScaleAtt. We use multiple heads ($H = 4$) like in the Multi-Head Attention (MHA) [23] formulation. For a single head, attention outputs are

$$\text{ScaleAtt} = \text{softmax}(\frac{qK^T}{\sqrt{d_k}})V, K = f_k(\mathbf{x}), V = f_v(\mathbf{x}). \quad (1)$$

q is a learnable query vector (per each head), which makes the formulation non self-attentive. K and V are key and value matrices obtained through f_k and f_v projection linear layers (output dimensions are $d_v = d_v = 256$). H parallel attention modules are utilized, and outputs are concatenated to form final d_{model} dimension output, where $d_{\text{model}} = H d_v$.

2.3. Speaker embedding networks

Light-ResNet34 [24]: LResNet [24] is a smaller ResNet-based x-vector architecture with 256-D embedding, 80-D Log-Mel FilterBank (LMFB) input features, and 5.6M parameters. It has four residual blocks whose outputs are used for DFL.
RawNet3 [25]: We utilize this 16.2M parameter time-domain model as authors showed it compatible with self-supervised techniques [25]. The first layer is a parametric analytic filterbank (256 filters) followed by three residual backbone blocks (1024 filters each). For DFL, we use the outputs of all convolutional blocks before the pooling layer.

2.4. Generative Adversarial Networks

For BWE model, we use Generative Adversarial Networks. We primarily focus on supervised GANs (Conditional GAN). Supervised Conditional GAN (CGAN) [26] learns to sample from

conditional distribution ($p_{A|B}$) where A and B are two domains (p_A, p_B resp.). Generator $\mathcal{G}_{A \to B}$ generates fake sample while discriminator D_B distinguishes between real and fake via

$$\max_{\mathcal{G}_{A \to B}} \min_{D_B} \mathcal{L}_{\text{CGAN}}, \quad \text{where} \quad \mathcal{L}_{\text{CGAN}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{sup}} \mathcal{L}_{\text{sup}}. \quad (2)$$

$\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{sup}}$ (weighted by λ_{sup}) are adversarial and supervised loss respectively:

$$\mathcal{L}_{\text{adv}}(\mathbf{a}, \mathbf{b}) = \mathbb{E}_{\mathbf{a},\mathbf{b} \sim p_{A,B}} [(\mathcal{D}(\mathbf{b}))^2 + (1 - \mathcal{D}(\mathcal{G}(\mathbf{a}))^2], \quad (3)$$

$$\mathcal{L}_{\text{sup}}(\mathbf{a}, \mathbf{b}) = \mathbb{E}_{\mathbf{a} \sim p_A, \mathbf{b} \sim p_B} [\|\mathbf{b} - \mathcal{G}_{A \to B}(\mathbf{a})\|_1]. \quad (4)$$

Here, $\mathcal{L}_{\text{adv}}$ is based on Least-squares GAN [27]. $\mathbf{a}$ and $\mathbf{b}$ are real paired samples from domains A and B respectively.

CycleGAN is an unpaired model using two tied CGANs:

$$\max_{\mathcal{G}_{A \to B}, \mathcal{G}_{B \to A}} \min_{\mathcal{D}_A, \mathcal{D}_B} \mathcal{L}_{\text{cyc-GAN}} \quad , \text{where} \quad (5)$$

$$\mathcal{L}_{\text{cyc-GAN}} = \mathcal{L}_{\text{adv}, A \to B} + \mathcal{L}_{\text{adv}, B \to A} + \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}} + \lambda_{\text{id}} \mathcal{L}_{\text{id}}. \quad (6)$$

Adversarial losses are defined like in Eq. 3. $\mathbf{a}$ and $\mathbf{b}$ are real unpaired samples. λ_{cyc} and λ_{id} are the weights for cycle and identity loss which are used for semantic consistency and regularization, respectively:

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{\mathbf{a} \sim p_A, \mathbf{b} \sim p_B} [\|\mathbf{a} - \mathcal{G}_{B \to A}(\mathcal{G}_{A \to B}(\mathbf{a}))\|_1]$$
$$+ \mathbb{E}_{\mathbf{a} \sim p_A, \mathbf{b} \sim p_B} [\|\mathbf{b} - \mathcal{G}_{A \to B}(\mathcal{G}_{B \to A}(\mathbf{b}))\|_1], \quad (7)$$

$$\mathcal{L}_{\text{id}} = \mathbb{E}_{\mathbf{a}, \mathbf{b} \sim p_{A,B}} [\|\mathbf{a} - \mathcal{G}_{B \to A}(\mathbf{a})\|_1 + \|\mathbf{b} - \mathcal{G}_{A \to B}(\mathbf{b})\|_1].$$

2.5. Generator and Discriminator architectures

For Generator, we use Conv-TasNet [28] (1.6M parameters). It consists of *encoder*, *separator*, and *decoder*. The *separator* has eight CNN layers that compute a mask. For separator, the kernel size is 16, the stride is 8, and channels increase from 128 to 1024 exponentially with a dilation of 2. For CGAN discriminator, we use Parallel WaveGAN (0.16M parameters) [29]. It is a 10-layer 1-D CNN with a kernel of 3, channels of 80, and linearly increasing dilation from the second to the ninth layer (from one to eight). For CycleGAN discriminator, we use multiperiod HiFiGAN (7M parameters) [30]. Each of the four subdiscriminator consists of six 1-D CNN layers with kernel size as 5, stride as 3, output channels as (4,16,64,256,1024,1).

Table 1: *Effect of different pooling methods and SSL model in Self-FiLM on ASV metrics: EER/minDCF-formatted (lower the better). We use pre-trained SSL and do not investigate pre-extension block and deep feature loss (Fig. 1). * denotes identical systems.*

CGAN BWE type	CGAN Sup loss	G+D params (M)	SRE16-YUE-eval40	SRE-CTS-superset-dev	SRE21-audio-eval
No BWE	-	-	7.12 / 0.376	5.36 / 0.216	17.12 / 0.644
BWE without self-FiLM	0.0048	1.7	5.68 / 0.332	4.08 / 0.180	15.81 / 0.618
Pooling type (SSL = wav2vec 2.0)					
mean	0.0049	5.7	5.41 / 0.306	4.05 / 0.180	15.11 / 0.610
mean+std	0.0051	7.6	7.09 / 0.335	4.01 / 0.178	13.84 / **0.586**
LDE	0.0052	18.2	**5.07 / 0.298**	**3.86 / 0.175**	14.33 / 0.593
ScaleAtt (*)	0.0052	39.3	5.22 / 0.303	4.03 / 0.181	14.09 / 0.591
SSL type (Pooling = ScaleAtt)					
wav2vec 2.0 (*)	0.0052	39.3	5.22 / 0.303	4.03 / 0.181	14.09 / 0.591
Robust Large wav2vec 2.0	**0.0035**	49.2	5.44 / 0.307	4.00 / 0.177	15.42 / 0.614
XLSR-53	**0.0035**	49.2	5.52 / 0.306	3.96 / 0.177	15.32 / 0.611
WavLM	0.0036	39.3	5.51 / 0.302	3.98 / 0.177	15.05 / 0.603
data2vec 2.0	**0.0035**	39.3	5.48 / 0.308	**3.94 / 0.175**	16.07 / 0.621

Algorithm 1 Steps in Self-FiLM (for CGAN BWE)

1. Pre-train an x-vector model and freeze it.
2. Pre-train an SSL model and freeze it.
3. (optional) Pre-train a simple BWE model ("prior BWE model" in Fig. 1) using regression or GAN loss and freeze it.
4. Arrange the above models per Fig. 1 configuration. Use x-vector as the auxiliary model in deep feature loss.
5. Using Eq. 3, train CGAN ("main BWE model") along with pooling and FiLM layers corresponding to $\mathcal{G}$ and $\mathcal{D}$.
6. During inference, discard the discriminator and feed the output of the "main BWE model" to the downstream ASV pipeline.

2.6. Feature-wise Linear Modulation

For data index i and channel index c, Feature-wise Linear Modulation (FiLM) operation/layer [4] adaptively modulates the activations $F_{i,c}$ of a Deep Neural Network (DNN) with a conditioning vector s_i. It learns linear layers (f_c, h_c, g_c) per channel. After introducing FiLM strength hyper-parameter $\alpha \in [0, 1]$,

$$\gamma_{i,c} = f_c(g_c(s_i)), \beta_{i,c} = h_c(g_c(s_i)),$$
$$\text{FiLM}(F_{i,c}) = F_{i,c} + \alpha(\gamma_{i,c}F_{i,c} + \beta_{i,c} - F_{i,c}). \quad (8)$$

Output dimension of f_c and h_c equals $F_{\cdot,c}$ channel dimension. Dimension of $s_{\cdot}$ depends on the pooling function, so we introduce g_c linear layer with a fixed output dimension of 256.

3. Self-FiLM

In Self-FiLM, we condition the layers of a DNN with the self-supervised representations of the input signal *itself*. It provides an alternate rich *view* of the signal. Our flexible framework allows clean integration of SSL models with existing pipelines and avoids system re-design. The test signal may benefit from SSL conditioning (a proxy for acoustic environment embedding) for zero-shot generalization during inference. In Fig. 1, we show a diagram of Self-FiLM on Conditional GAN (both generator and discriminator). Narrowband signal x_n is input to an optional preliminary BWE model (*pre-extension*) for improved compatibility with the (usually) wideband-trained SSL model. The self-supervised representations are pooled (Sec. 2.2) and FiLM-ed to all convolutional layers of the generator and the discriminator. We use the modified FiLM proposed in Eq. 8. Our initial methodology involves leveraging a pre-trained publicly available SSL model to obtain a sequence of self-supervised embeddings $(s_1, \ldots, s_T)$ where T is the number of such vectors. We use the last layer of SSL models for

simplicity. For CGAN optimization, we also explore using an auxiliary speaker embedding network for deep feature loss [31] to preserve speaker identity during BWE. Algorithm 1 summarizes the steps. Self-FiLM can be seen as top-down conditioning, while DFL can be seen as bottom-up conditioning. Self-FiLM CGAN is thus a semi-supervised model, while the Cycle-GAN version is entirely unsupervised. Our preliminary work on applying Self-FiLM to x-vector led to over-fitting on in-domain data, so we defer that to future work.

4. Experimental Setup

As stated in Sec. 1, we aim to improve ASV performance on telephone test sets using 16KHz SSL models. Richer wideband training data is available to train x-vector (LResNet, RawNet3) and SSL models. X-vector and Probabilistic Linear Discriminant Analysis (PLDA) are trained on VoxCeleb [32], narrowband VoxCeleb (*VoxCeleb_narrow*), and Speaker Recognition Evaluation (SRE) telephone data (*SRE_telephone*). VoxCeleb (1&2 combined) contains 2700+ hrs of audio from 7365 speakers in the wild. *VoxCeleb_narrow* is created by removing upperband information (4-8KHz). *SRE_telephone* (11k h, 6909 speakers) is created by combining SRE Superset [33] and SRE16 eval data [10] which includes Tagalog and Cantonese (YUE) languages. In GAN, G and D are initialized randomly and VoxCeleb_narrow and VoxCeleb are used for domains A and B data. We test on three sets that cover a variety of languages, and acoustic environments: *SRE16-YUE-eval40* (40% speakers (40) from evaluation set of SRE16 Cantonese), *SRE-CTS-superset-dev* (99 speakers from CMN (Mandarin) and YUE (Cantonese) languages, 6M trials), and *SRE21-audio-eval* [10,34] (contains various languages and domains, 22M trials). Test set details may be found in previous works [10, 34]. Note that the narrowband signals are also resampled from native 8KHz to 16KHz. During testing, BWE is applied to all test signals (including the wideband signals of SRE21) and PLDA narrowband portion. LResNet34, RawNet3 are obtained pre-trained from prior works [24, 25] on VoxCeleb, VoxCeleb_narrow, and SRE_telephone with Additive Angular Margin (AAM) softmax (margin=0.3) speaker classification loss and data augmentation (MUSAN noises, Aachen Room Impulse Response reverberations) [21]. Training configurations for GANs are obtained from [10]. For ASV evaluation, we use Equal Error Rate (EER) and minimum Decision Cost Function (minDCF) metrics (with a target speaker prior probability of 0.05) which capture false positive and false negatives. LResNet-PLDA pipeline is used for scoring [24]. All models are trained with PyTorch with 4x24GiB GPUs.
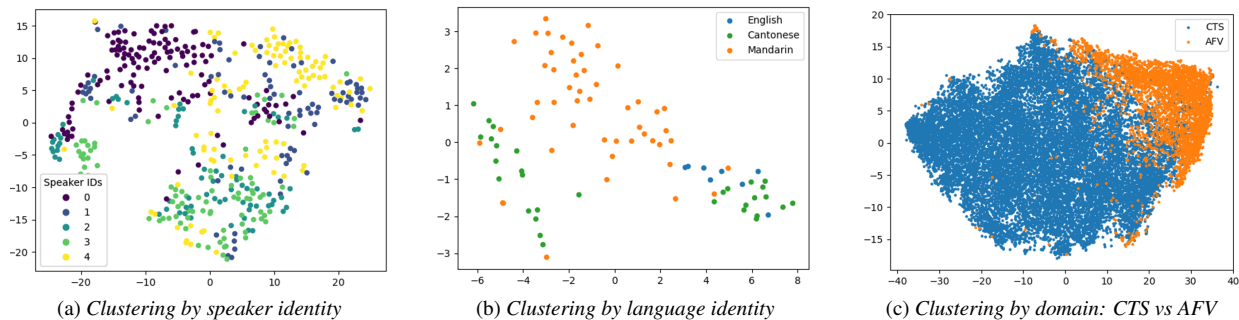
| (a) *Clustering by speaker identity* | (b) *Clustering by language identity* | (c) *Clustering by domain: CTS vs AFV* |

Figure 2: *t-SNE visualization of FiLM activations from the first layer of CGAN generator using SRE21 test data.*

5. Results

5.1. Exploring pooling methods and SSL models

Here, we show the effectiveness of Self-FiLM under a wide variety of pooling methods and SSL model choices (Table 1). First row contains the baseline results without BWE. With a CGAN BWE, we can see drastic improvement across all test sets and obtain a supervision loss value of 0.0048. This recreates results from previous studies [1, 10] and establishes a strong baseline for further experiments as the CGAN has been tuned extensively. Next, we investigate pooling methods using wav2vec 2.0 Self-FiLM. Here, we see higher improvements with complex methods like mean+std, LDE, and ScaleAtt. Note that BASE wav2vec 2.0 does not improve CGAN supervision loss. Then, we investigate stronger SSL models with ScaleAtt pooling, which all improve the baseline. We note that the results do not necessarily improve consistently with larger models as corroborated by prior studies [5]. However, lower supervision loss and the observed improved GAN training reveal the potential of such models. We visualize that Self-FiLM can extract discriminative information about the speaker, language, and domain from SSL models (Fig. 2). We apply t-SNE on FiLM activations from the first layer of the CGAN's generator. In Fig. 2(a), we plot recordings of five random speakers and observe clustering by speaker identity. In Fig. 2(b), we fix a speaker and discover clustering by the languages spoken. In Fig. 2(c), we observe clustering by telephone (CTS) and wideband (AFV) domains.

Table 2: *Investigating role of in-domain SSL model, α, pre-extension, deep feature supervision loss in Self-FiLM CGAN.*

BWE type	SRE16-YUE	SRE-CTS	SRE21
No BWE	7.12 / 0.376	5.36 / 0.216	17.12 / 0.644
Training data for data2vec 2.0			
Librispeech	5.48 / 0.308	**3.94 / 0.175**	16.07 / 0.621
VoxCeleb	5.52 / 0.305	3.98 / 0.176	15.09 / 0.604
SRE_telephone	5.40 / 0.307	3.97 / 0.176	15.45 / 0.608
SRE_telephone+VoxCeleb8k	5.56 / 0.308	3.99 / 0.177	14.81 / 0.598
VoxCeleb data2vec 2.0	5.52 / 0.305	3.98 / 0.176	15.09 / 0.604
+ α=0.5	5.59 / 0.316	4.02 / 0.178	14.59 / 0.595
+ pre-extension	5.43 / 0.303	3.95 / 0.176	15.57 / 0.612
+ feature-domain DFL	5.51 / 0.305	3.97 / 0.176	15.01 / 0.605
+ time-domain DFL	5.42 / **0.301**	4.00 / 0.178	**14.07** / 0.589
+ α=0.5+time DFL	5.40 / 0.305	4.01 / 0.178	14.08 / **0.588**
+ α=0.5+time DFL+pre-extend	**5.28** / 0.308	3.98 / 0.180	14.20 / 0.591

5.2. Effect of in-domain training data for data2vec 2.0, FiLM strength α, pre-extension and deep feature loss

We choose data2vec2.0 for further analysis (Table 2) as it has the highest training efficiency [19]. Test set names are shortened for brevity. First, we train data2vec 2.0 on different datasets.

We find training on naturalistic wideband and narrowband data to be better than out-of-domain (OOD) read speech corpus Librispeech. We get even better results than Robust wav2vec 2.0 (Table 1). We combined VoxCeleb_narrow and SRE_telephone to observe great performance, but it led to unstable GANs for further experiments with DFL. This is perhaps because the existing training configuration of data2vec 2.0 is optimized for wideband data. Using VoxCeleb data2vec 2.0, we conduct further experiments. Using $\alpha = 0.5$ on CGAN gives slight improvement, while bandwidth *pre-extension* gives slightly inconsistent gains. With deep feature loss in CGAN training, we observe significant improvements, especially with the temporal model (RawNet3), as CGAN operates in the time domain. Finally, we try combinations of the above experiments. We observe synergy in different test sets which can be advantageous for ASV *fusion* [34].

5.3. Application to CycleGAN-based bandwidth extension

Here, we prove the compatibility of unsupervised BWE models based on CycleGAN with Self-FiLM. In Table 3, we note the benefit of using (default) Librispeech and VoxCeleb data2vec2.0. Performance on SRE21 is greatly improved while other test sets benefit from RawNet3 based DFL (in cycle and identity loss). On SRE21, there is degradation in the last row perhaps due to 1) usage of CGAN hyper-parameter configuration or 2) using identical RawNet3 for the other generator, which learns the reverse mapping. In the future, we can utilize a larger computational budget to explore ideal CycleGAN hyper-parameters and training data configurations. We can also explore joint training of CycleGAN and SSL in the future.

Table 3: *Integration of unsupervised BWE with Self-FiLM*

BWE type	SRE16-YUE	SRE-CTS	SRE21
No BWE	7.12 / 0.376	5.36 / 0.216	17.12 / 0.644
BWE w/o Self-FiLM	**4.95** / 0.294	3.99 / **0.176**	17.58 / 0.681
Librispeech Self-FiLM	4.97 / 0.297	4.16 / 0.183	15.96 / 0.637
VoxCeleb Self-FiLM	5.47 / 0.317	4.52 / 0.197	**14.02 / 0.609**
+ time-domain DFL	5.09 / **0.290**	**3.98** / 0.180	16.75 / 0.637

6. Conclusion

We proposed *Self-FiLM* to condition a BWE model with the self-supervised representation of the input signal itself. We also corroborate the findings of Robust wav2vec 2.0 by training data2vec 2.0 on mixed-bandwidth in-domain data. In our framework, we showed data2vec 2.0 is compatible with narrowband inputs, prior BWE (*pre-extension*) model, and deep feature loss-based BWE. Finally, we extend Self-FiLM to combine CycleGAN and data2vec 2.0 for a fully unsupervised solution.

7. References

[1] S. Kataria, J. Villalba, L. Moro-Velázquez, and N. Dehak, "Joint domain adaptation and speech bandwidth extension using time-domain gans for speaker verification," *arXiv preprint arXiv:2203.16614*, 2022.

[2] S. Han and J. Lee, "Nu-wave 2: A general neural audio upsampling model for various sampling rates," *arXiv preprint arXiv:2206.08545*, 2022.

[3] B. Gfeller, D. Roblek, and M. Tagliasacchi, "One-shot conditional audio filtering of arbitrary sounds," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 501–505.

[4] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[5] H. Song, S. Chen, Z. Chen, Y. Wu, T. Yoshioka, M. Tang, J. W. Shin, and S. Liu, "Exploring wavlm on speech enhancement," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 451–457.

[6] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[7] V.-A. Nguyen, A. H. Nguyen, and A. W. Khong, "Tunet: A block-online bandwidth extension model based on transformers and self-supervised pretraining," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 161–165.

[8] V. N. Sukhadia, A. Arunkumar, and S. Umesh, "Channel-aware pretraining of joint encoder-decoder self-supervised model for telephonic-speech asr," *arXiv preprint arXiv:2211.01669*, 2022.

[9] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[10] S. Kataria, J. Villalba, L. Moro-Velázquez, P. Żelasko, and N. Dehak, "Time-domain speech super-resolution with gan based modeling for telephony speaker verification," *arXiv preprint arXiv:2209.01702*, 2022.

[11] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.

[12] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve *et al.*, "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pretraining," *arXiv preprint arXiv:2104.01027*, 2021.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[15] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[16] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[17] M. J. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low-resource languages: Babel project research at cued," in *Fourth International workshop on spoken language technologies for under-resourced languages (SLTU-2014)*. International Speech Communication Association (ISCA), 2014, pp. 16–23.

[18] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[19] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[20] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[21] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin *et al.*, "The jhu-mit system description for nist sre18," *Johns Hopkins University, Baltimore, MD, Tech. Rep*, 2018.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," *arXiv preprint arXiv:1906.09890*, 2019.

[24] J. Villalba, D. Garcia-Romero, N. Chen, G. Sell, J. Borgstrom, A. McCree, L. Garcia-Perera, S. Kataria, P. S. Nidadavolu, P. A. Torres-Carrasquillo *et al.*, "Advances in speaker recognition for telephone and audio-visual data: the jhu-mit submission for nist sre19," in *Proceedings of Odyssey*, 2020.

[25] J.-w. Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," *arXiv preprint*, vol. 2203, 2022.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[28] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[29] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[30] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[31] S. Kataria, P. S. Nidadavolu, J. Villalba, and N. Dehak, "Analysis of deep feature loss based enhancement for speaker verification," *arXiv preprint arXiv:2002.00139*, 2020.

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

[33] O. Sadjadi, C. Greenberg, E. Singer, L. Mason, D. Reynolds *et al.*, "Nist 2021 speaker recognition evaluation plan," 2021.

[34] J. Villalba, B. J. Borgstrom, S. Kataria, M. Rybicka, C. D. Castillo, J. Cho, L. P. Garcıa-Perera, P. A. Torres-Carrasquillo, and N. Dehak, "Advances in cross-lingual and cross-source audio-visual speaker recognition: The jhu-mit system for nist sre21," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 213–220.