IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 6, 2022, accepted May 31, 2022, date of publication June 8, 2022, date of current version June 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3181298

Length-Normalized Representation Learning for Speech Signals

KYUNGGUEN BYUN, SEYUN UM, AND HONG-GOO KANG, (Member, IEEE)
Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Hong-Goo Kang (hgkang@yonsei.ac.kr)

This research was supported by the Yonsei Signature Research Cluster Program of 2022 (2022-22-0002)

ABSTRACT In this study, we proposed a length-normalized representation learning method for speech and text to address the inherent problem of sequence-to-sequence models when the input and output sequences exhibit different lengths. To this end, the representations were constrained to a fixed-length shape by including length normalization and de-normalization processes in the pre- and post-network architecture of the transformer-based self-supervised learning framework. Consequently, this enabled the direct modelling of the relationships between sequences with different length without attention or recurrent network between representation domains. This method not only achieved the aforementioned regularized length effect but also achieved a data augmentation effect that effectively handled differently time-scaled input features. The performance of the proposed length-normalized representations on downstream tasks for speaker and phoneme recognition was investigated to verify the effectiveness of this method over conventional representation methods. In addition, to demonstrate the applicability of the proposed representation method to sequence-to-sequence modeling, a unified speech recognition and text-to-speech (TTS) system was developed. The unified system achieved a high accuracy on a frame-wise phoneme prediction and exhibited a promising potential for the generation of high-quality synthesized speech signals on the TTS.

INDEX TERMS Self-supervised learning, representation learning, speech and text analysis.

I. INTRODUCTION

Deep learning systems trained with a sufficient amount of data with accurate labels using supervised learning exhibit a high performance [1]–[3]. However, owing to the high cost required to obtain accurately labeled data, semi-supervised learning approaches have emerged as suitable alternatives for cases with a limited amount of labeled data [4]. These approaches extract latent space representations without utilizing external label information in the pre-training stage, after which the target task with a small amount of labeled data is subjected to a fine-tuning process. Nevertheless, either unsupervised [5]–[7] or semi-supervised [8]–[11] learning approaches can be used in the pre-training.

Recently, self-supervised learning approaches that extract latent representations by training on the relationship between data and data-oriented supervision have attracted widespread attention [12], [13] because of their ability to extract high quality features without using label. An example of this approach is the modification of input data using pre-defined transforms or modifications, after which the network is trained to estimate or recover the original input.

A previous study employed a prediction-based self-supervision [14] method that utilizes the time–domain correlation of nearby samples to obtain representations for sequential data, such as speech and audio. This prediction method exhibited a significantly higher performance than conventional signal processing-based features in various voice interface applications [14]–[16]. In addition, another study [17] applied vector quantization to speech representations to obtain discretized outputs similar to word tokens in natural language processing.

Furthermore, the use of an encoder–decoder network with reconstruction loss has been utilized to obtain representations for sequential data. In addition, bidirectional encoder representations from transformers (BERT) [18], which obtains the latent space representations of word sequences while training under an auto-encoder framework, has been adopted for extracting speech representation features from spectrum inputs. Examples include MockingJay [19], audio ALBERT (AAL-BERT) [20], and transformer encoder representation from alteration (TERA) [21]. Similar to the cloze task in

The associate editor coordinating the review of this manuscript and approving it for publication was Taous Meriem Laleg-Kirati.

BERT, MockingJay and AALBERT initially distort an input spectrum using time–domain random masks, after which the models are trained to reconstruct the original input spectrum. In contrast, TERA [21] extends the time–domain random masking technique to a frequency domain to provide more diverse variations to the input spectrum, which enables a deeper learning of internal representations.

However, the difficulties in modeling the dynamic or time-varying mapping relationship between the sequential data and its label information has remained a challenging issue for sequence-based applications (e.g., speech applications). For example, the duration required for the recognition of each phoneme/word in speech recognition varies depending on the type of phoneme, speaker, and speaking style. To solve this problem, a previous study proposed a sequence-to-sequence (S2S) model with attention [22]; however, the S2S model exhibited a slow inference speed owing to its auto-regressive generation process in the decoding step. This is because when the length of the output sequence is unknown, the model needs to determine which part of the input embedding to focus on at each decoding step; thus, resulting in a slow inference speed.

In this study, we investigated the application of a self-supervised learning method for obtaining regularized fixed-shaped latent representations from sequential data of variable lengths, particularly, for speech and text data. To this end, an arithmetic re-sampling process was integrated into, before, and after a conventional transformer-based representation learning method. This ensured that the length of the representations remained constant regardless of the length of the input signal without additional network parameters.

We also investigated the effect of a length-normalization process on various aspects of the learned representations, including the performance of the process on reconstruction task and other downstream tasks, such as speaker recognition and phoneme recognition. The appropriate length of the latent representation was determined by investigating the characteristics of the database. Lastly, we proposed a system that can reliably perform speech recognition and text-to-speech (TTS) tasks within a unified framework using the proposed representation method to demonstrate the strength of the method. The proposed method was expected to exhibit potential application to most transformation systems that utilize S2S modeling architectures, including, speaking style modification, and audio-visual applications, such as lip-to-speech.

The major contributions of this study are as follows; i) we proposed a versatile length-normalized (LN) self-supervised learning based representation extraction method that can encode any arbitrary length sequences into vectors of a constant shape. The effectiveness of this method was verified by evaluating the performance of the learned features on reconstruction and downstream tasks; ii) we proposed a fully feed-forward S2S framework, which utilized the proposed LN representation that overcomes the auto-regressive decoding process normally observed in S2S problems.

II. TRANSFORMER-BASED SELF-SUPERVISED LEARNING METHODS FOR SPEECH DATA

MockingJay [19] is a transformer-based speech representation learning method that utilizes frame-wise speech features, such as mel-filterbank energies and mel-frequency cepstral coefficients (MFCCs), as tokens for the BERT model. This method adopts a self-supervised learning framework that utilizes ground truth speech features as label information and masked features as inputs of the model. The model parameters were trained to minimize the reconstruction loss.

Fig. 1 shows a schematic illustration of the overall block diagram of the MockingJay architecture. First, the entire frequency region of the time–domain feature frames were randomly masked. After the masking process, positional encoding was added using simple sinusoidal functions. In addition, a multi-head attention layer was utilized to capture the internal relationships within the input sequence. Furthermore, the attention scores were calculated parallelly using the scaled dot-product attention method, which was achieved by dividing the input into queries, keys, and values depending on the number of heads [22], after which the scores were concatenated.

Subsequently, the final representation was obtained by passing the output of the multi-head attention layer through the activation and intermediate feed-forward layers. In the decoding process, the original spectrum was reconstructed from the obtained representations using feedforward layers. Speaker and phoneme recognition tasks were performed using MockingJay to demonstrate the effectiveness of the learned representations compared to conventional acoustic features, such as MFCCs and mel-filter bank energies [19]. Based on the MockingJay framework, AALBERT [20] utilizes the shared parameters across the transformer layers to reduce the number of parameters, while obtaining a similar performance as MockingJay on downstream tasks.

The aforementioned transformer-based self-supervised learning methods exhibited good pre-training performance using the learned representations on various downstream tasks. However, as these methods produced variable length representations depending on the length of the input sequence, an auto-regressive decoding process is required in an attention-based encoder–decoder framework. To implement a parallel decoding process in S2S applications, we proposed a method that ensures that the shape of the representation remains constant regardless of the length of the input sequence. To implement a parallel decoding process in S2S applications, the shape of the representation should remain constant regardless of the length of the input sequence. We proposed a method that ensures this.

III. METHOD AND APPLICATION OF LENGTH-NORMALIZED (LN) REPRESENTATIONS

A. METHOD

Fig. 2 shows the structure of the proposed LN representation model. Similar to the MockingJay model, an alteration

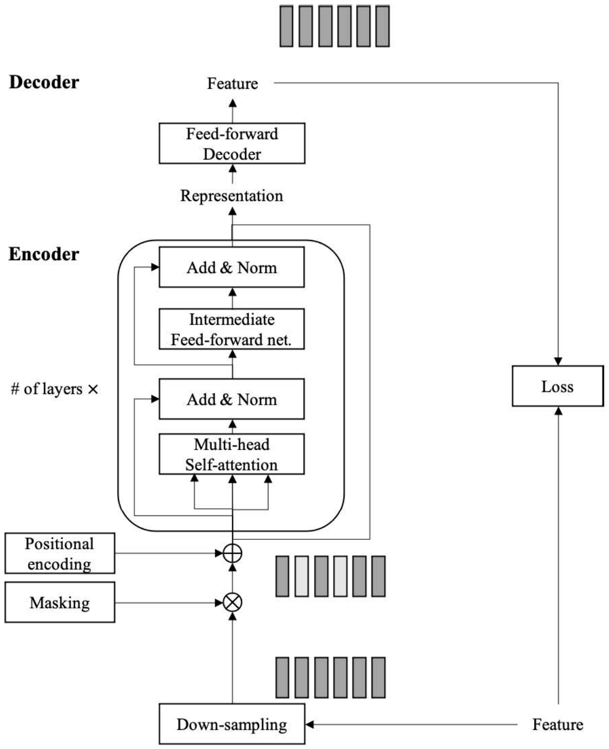


FIGURE 1. **Overall block diagram of MockingJay.**

process, which is a random masking technique in the time domain, was applied to the input features. Furthermore, conventional features, such as mel-filterbank energies and MFCCs, were utilized for the input feature of the representation model. In addition, positional information was provided to the transformer layers of the LN representation by adding pre-computed sinusoidal values to the input.

Subsequently, the position embedded feature was re-sampled to a pre-defined length (hereafter denoted as the time–axis dimension) using an algebraic interpolation method as neural network-based up-sampling or down-sampling layers can only change the sampling rate with a fixed ratio. It is important to note that no additional network parameters are required for the re-sampling process.

After the re-sampling process, the re-sampled features and its ratios were encoded into a fixed-shaped representation using the transformer layers. In addition, the self-attention module inside the transformer enabled the modeling of the internal relationships between input frames without recurrent network.

The extracted representation was decoded using fully connected feed-forward layers and was re-sampled to achieve the original length using the inverse of the encoded re-sampling ratio. Lastly, the entire network was trained to minimize the distance between the ground truth feature and the reconstructed feature from the representation model using:

$$\mathcal{L}_{rec} = ||\mathbb{D}_X(\mathbb{E}_X(M \cdot X)) - X||_1, \tag{1}$$

where M is the masking matrix that is randomly generated in every sample, X is the two-dimensional input feature, such as a mel-spectrogram, and $\mathbb{E}_X$, and $\mathbb{D}_X$ are the encoder and decoder respectively. It is important to that the length of the reconstructed feature was adjusted to the original input length before the loss was calculated.

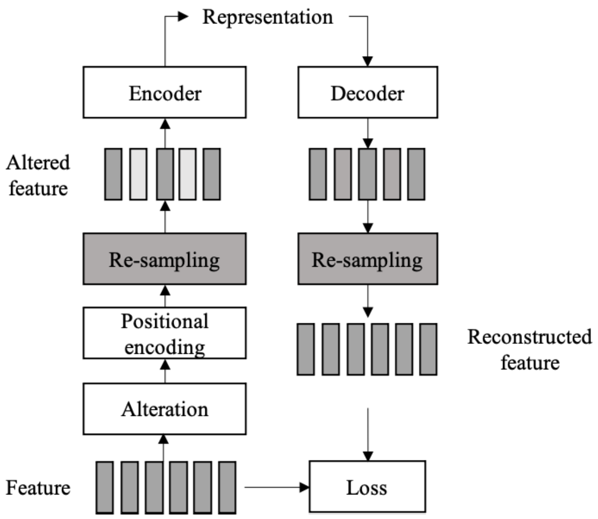


FIGURE 2. **Proposed length-normalized representation learning model.**

Compared to conventional transformer-based self-supervised learning methods, the method proposed in this study has two advantages. First, owing to the fixed length of the representation in each domain, only a feed-forward network architecture (i.e., without using a recurrent architecture) was required to utilize the proposed representation in S2S applications. Second, similarly to data augmentation, this method automatically generalizes the capabilities of the learned representations because it handles various sets of time-scaled information owing to the re-sampling operation.

B. APPLICATION: TRANSFORMATION BETWEEN SPEECH SPECTRUM AND PHONEMES

To investigate the practicability of the LN representation extraction method proposed in this study, the method was applied to automatic speech recognition (ASR) and TTS applications, which are two of the most common S2S problems in the joint speech and text domains. Several studies have investigated the combination of ASR and TTS systems [23], [24]. These unified systems are useful when labeled data are unavailable as pseudo-labeled information may be utilized by predicting them using ASR and TTS systems. To convert representations from one domain to the other, two auto-encoder and S2S models for speech and text domains are required. However, S2S processing leads to an auto-regressive decoding process, which decreases the prediction speed; thus, significantly increasing the computational complexity. On the contrary to the conventional algorithms, in this study, we demonstrated that the LN representation can be applied to ASR and TTS predictions in a feed-forward manner.

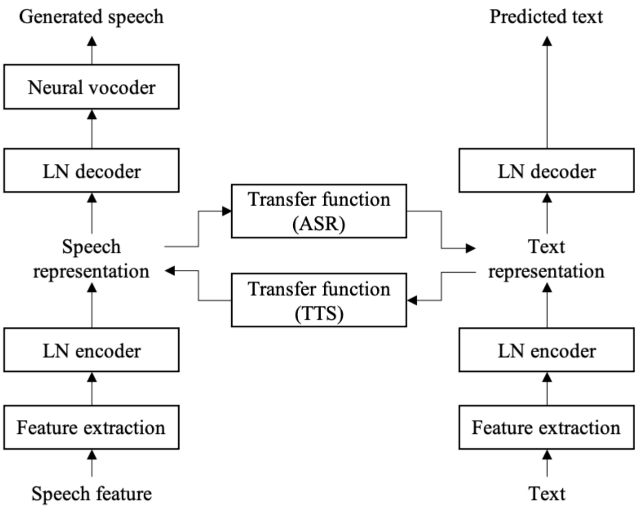


FIGURE 3. Unified ASR-TTS framework using the proposed representation learning model that does not need a cross attention mechanism.

Fig. 3 shows the structure of the unified system that utilized the proposed representation learning method as a baseline framework. The unified system consisted of two independent LN representation learning models for speech and text sequences. The speech and text representations were extracted from the encoder of each model. First, the LN representation model was independently trained on speech and text, after which a CycleGAN [25]-based conversion model was utilized for transformation: speech representations and text representations were utilized as the inputs and outputs of the ASR system, respectively, and vice versa for the TTS system. To model the text representation, frame-wise phoneme index sequences were converted into a two-dimensional encoded one-hot text embedding using the transcription information obtained from Montreal Forced Aligner [26]. Subsequently, the text feature was used as the input in the LN text representation model. An example of the text feature is shown in Appendix.

For the ASR system, speech was encoded into LN speech representations and converted to text representations. Subsequently, the resulting text representations were decoded into text or phoneme sequences. For the TTS system, text was converted to text features using the transcription information and encoded into LN text representations. Subsequently, speech representations were generated and decoded into spectral features using the LN speech decoder. Lastly, speech waveforms were synthesized using a neural vocoder. As all the prediction processes do not utilize recurrent connections, predictions were performed parallelly.

Some previous studies have investigated the fusion of representations from different domains into the same embedding space [27]–[29]. In addition to the elaborately designed fusion method, a unified embedding space of visual-text or speech-text information was proposed to improve the performance of the representation. However, the fusing approach for audio and text representations was not considered in this model. This is because although the fusing approach may obtain optimal representations between two domains, an independent representation learning simplifies the training process and facilitates multi-domain learning by dividing the process into a representation learning and transformation between representation parts.

If X_s and Y_t represent speech and text representations, respectively, and the proposed framework consisted of a speech representation generator (G_X) and a text representation generator (G_Y), and D_X and D_Y corresponds to the discriminators for speech and text representation vectors, respectively, the entire loss function of the transformation model can be represented using:

$$\mathcal{L} = (1-\lambda_{pred})\mathcal{L}_{adv} + \lambda_{pred}(\mathcal{L}_{cyc} + \mathcal{L}_{pred} + \mathcal{L}_{dec}), \quad (2)$$

where $\mathcal{L}_{adv}$ and $\mathcal{L}_{cyc}$ are the adversarial loss and cycle consistency loss that are typically used for training CycleGAN, respectively. $\mathcal{L}_{pred}$ and $\mathcal{L}_{dec}$ are additional proposed loss terms for enhancing the performance of the model by minimizing the prediction error terms in the representation and decoded sequence domains, respectively. λ_{pred} controls the importance of the prediction error terms and adversarial loss (it begins with a small value at the initial stages of training and increases as the training progresses).

The adversarial loss was defined using a zero-sum game between two competitive networks (i.e., generator and discriminator) [30]. The generator attempts to generate network outputs that can deceive the discriminator, whereas the discriminator attempts to classify the given input as a ground truth or a generated fake input. As the proposed model had two generator and discriminator pairs, the adversarial loss for each domain can be represented as follows:

$$\begin{aligned}\mathcal{L}_{adv} &= \mathcal{L}_{adv_X} + \mathcal{L}_{adv_Y}, && (3)\\ \mathcal{L}_{adv_X} &= |D_X(G_X(Y_t)) - 1| + |D_X(X_s)|, && (4)\\ \mathcal{L}_{adv_Y} &= |D_Y(G_Y(X_s))) - 1| + |D_Y(Y_t)|. && (5)\end{aligned}$$

In addition, the feature matching loss term that minimizes the difference between the intermediate layer outputs of the discriminators was used for the training.

The cycle consistency loss measured the difference between the ground truth input and the re-generated input after two transformations. The cycle consistency loss can be mathematically expressed using:

$$\begin{aligned}\mathcal{L}_{cyc} &= \mathcal{L}_{cyc_X} + \mathcal{L}_{cyc_Y}, && (6)\\ \mathcal{L}_{cyc_X} &= |G_X(G_Y(X_s)) - X_s|, && (7)\\ \mathcal{L}_{cyc_Y} &= |G_Y(G_X(Y_t)) - Y_t|. && (8)\end{aligned}$$

To further improve the performance of the generator, we proposed an additional prediction loss term compared to vanilla CycleGAN. This loss term minimized the prediction error between the ground truth and generator output in the representation domain as follows:

$$\begin{aligned}\mathcal{L}_{pred} &= \mathcal{L}_{pred_X} + \mathcal{L}_{pred_Y}, &(9)\\ \mathcal{L}_{pred_X} &= |G_X(Y_t) - X_s|, &(10)\\ \mathcal{L}_{pred_Y} &= |G_Y(X_s) - Y_t|. &(11)\end{aligned}$$

Similarly to the prediction loss, we propose a decoder loss to reduce the error in the decoded feature domains of the speech and text sequences. This minimized the error between decoded sequences from the ground truth representations and those of the generated representations as follows:

$$\begin{aligned}\mathcal{L}_{dec} &= \mathcal{L}_{dec_X} + \mathcal{L}_{dec_Y}, &(12)\\ \mathcal{L}_{dec_X} &= |\mathbb{D}_X(X_s) - \mathbb{D}_X(G_X(Y_t))|, &(13)\\ \mathcal{L}_{dec_Y} &= |\mathbb{D}_Y(Y_t) - \mathbb{D}_Y(G_Y(X_s))|, &(14)\end{aligned}$$

where $\mathbb{D}_X$ and $\mathbb{D}_Y$ are the decoders of the LN representation models for speech and text, respectively. For the TTS task, a neural vocoder module was used to generate time domain waveforms from the generated spectral information. In addition, a pre-trained HiFi-GAN vocoder was utilized in this study [31].

IV. EXPERIMENTS

In this study, several experiments were performed to verify the performance of the proposed LN representation model. First, the reconstruction error was investigated by utilizing speech spectra as inputs. Subsequently, the obtained representations were applied to various downstream tasks to evaluate their effectiveness, after which the relationship between the time–axis dimension and performance was investigated. Lastly, the results of the fully feed-forward unified ASR and TTS system that utilized the proposed latent representations from speech and text database were provided.

A. EXPERIMENTAL SETTINGS

The performance of the proposed representation model was compared to those of conventional ones using LibriSpeech [32]. LibriSpeech is a publicly accessible large scale multi-speaker database consisting of 1,000 h of speech waveform sampled at 16 kHz, and the corresponding text transcriptions. To obtain representations using the self-supervised learning method, the entire *train-clean100* subset was used for the training. For the downstream tasks, we utilized the data split settings used to evaluate contrastive predictive coding (CPC) [16], which divides the *train-clean100* subset into train, development, and test sets at a ratio of 8:1:1.

Table 1 summarizes the hyper-parameter settings used for the representation modeling of speech signals. The representation learning models were trained by utilizing mel-filterbank energies as the input features, and the window length and shift length of the model were set to 25 and 10 ms, respectively. In addition, the MockingJay setting was utilized in the network architecture to ensure that the model focused on the effect of the proposed re-sampling processes. Furthermore, Adam [33], which exhibits a learning rate of 0.0002, was utilized as the optimizer, and a batch size of 10 was used for the training.

TABLE 1. Hyper-parameter settings for acoustic representation learner.

Input	Mel-filterbank	80, [0, 8000]	
Output	Mel-filterbank	80, [0, 8000]	
Pre-processing	Mask ratio	0.15	
	Down-sampling rate	3	
Network Architecture	Transformer Encoder	# of layers	3
		Hidden size	768
		# of heads	12
		Intermediate FC layer size	3072
	Decoder	3FC layers with norm	
Optimizer setting	Optimizer	Adam	
	Learning rate	0.0002	
	Batch size	10	

B. PERFORMANCE OF THE REPRESENTATION MODEL

Various experiments were conducted to investigate the effect of the time–axis dimension of the proposed model on its performance. In addition, the validity of the proposed LN representations was evaluated by comparing their reconstruction loss with those of the representations from MockingJay and AALBERT. Finally, the performance of the obtained representations on downstream tasks was compared to those of the two baseline methods to evaluate their capabilities.

1) EFFECT OF TIME-AXIS DIMENSION

The change in the performance of the method with a variation in the time–axis dimension was investigated. The setting utilized for the network architecture is described in Table 1. The distribution of the length of the sentences in the *train-clean-100* data is shown in Fig. 4. Considering the distribution of the data, the value of the time–axis dimension was adjusted between 192 and 1536 (approximately 1.9 and 15.4 s), while constraining the structure of the LN representation learning model.

Depend on the time-axis dimension setting and the length of the input, the proposed model differently operates in re-sampling blocks. In case of the samples whose length is smaller than time-axis dimension, up-sampling occurs at the first re-sampling block and down-sampling occurs at the second re-sampling block. Otherwise, re-sampling block behavior opposite way. To determine the effect of re-sampling blocks to reconstruction error, the utterances of the test set were divided into up-sampled and down-sampled utterances.

The reconstruction errors of the proposed model are summarized in Table 2. The results revealed that when the up-sampling conversion was applied after the down-sampling conversion, the reconstruction error for the "down-sampled" cases was significantly larger than that of the "up-sampled" cases when the down-sampling conversion was applied after up-sampling. This could be attributed to the fact that the information loss during the down-sampling process was larger than that of the up-sampling process. In addition,

TABLE 2. Change in the reconstruction errors with variations in the time-axis dimension.

		Time-axis dimension			
		192	384	768	1536
Recon. Loss	Total	0.891	0.575	0.380	0.252
	Up	0.320	0.288	0.247	0.245
	Down	0.892	0.588	0.401	0.290

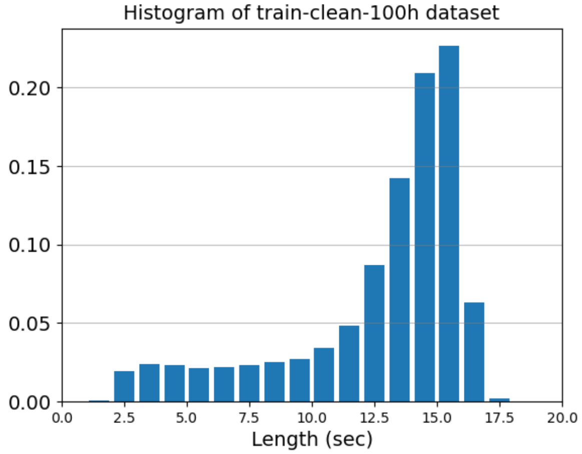


FIGURE 4. Histogram of the length of the *train-clean-100h* database (total 28539 sentences, minimum, maximum and mean length of sentences are 1.42, 24.53, and 12.69 s, respectively).

with an increase in the time–axis dimension, the ratio of the "up-sampled" sentences increased, resulting in a decrease in the overall reconstruction error.

2) COMPARISON OF THE RECONSTRUCTION PERFORMANCE

The detailed network specifications of the three models and their reconstruction losses when their training was converged are summarized in Table 3. The time–axis dimension of the LN models was set to 1536, which was determined by considering the distribution of the sentence lengths in the LibriSpeech database and the reconstruction loss. As the representation learning model was trained using a self-supervised method, all the data were used for training; however, sentences longer than 1600 frames were excluded. Among the models, MockingJay exhibited the lowest reconstruction loss, which could be attributed to the fact that it had the largest number of parameters and non-shared parameters in the transformer layers. In addition, the model proposed in this study not only showed comparable reconstruction performances to representation of MockingJay and AALBERT model but also extracted regularized latent features from the input signal. As previously stated, in contrast to the MockingJay and AALBERT models, the input sentence length had no effect on the representation vector size of the model proposed in this study. Examples of the input, ground truth, and reconstructed speech features are provided in Appendix.

TABLE 3. Reconstruction loss and model specifications of conventional models and the model proposed in this study.

	MockingJay	AALBERT	Proposed
Recon. loss (speech)	0.1824	0.2082	0.2519
Hidden size	768	768	768
Shared Layers	No	Yes	Yes
FF size	3072	3072	3072
Rep. size	(L/3, 768)	(L/3, 768)	(512, 768)
# of params.	22.23M	8.051M	8.051M

3) DOWNSTREAM TASK

a: EFFECT OF THE TIME–AXIS DIMENSION ON DOWNSTREAM TASKS

The effect of the time–axis dimension on the downstream tasks was investigated. For the experiments, 80 dimensional mel-filterbank energies were used as the input features. The speaker classification models were set as a single layer feed-forward network for all the cases using a Softmax function. The phoneme recognition task was performed using two simple feed-forward networks (i.e., a linear classifier and a feed-forward model with a single hidden layer). The LN process proposed in this study was not used in the frame-level inference of the phoneme recognition case to limit the amount of information and asynchronization between the frame-level labeled phoneme IDs. The results of the frame-level speaker recognition and phoneme recognition using a linear classifier and single hidden layer classifier are summarized in Table 4.

The experimental results revealed that there was a trade-off relationship between the accuracy of the speaker and phoneme recognition with a change in the time–axis dimension of the upstream models. As the time–axis dimension increases, the phoneme recognition accuracy increases. However, when the time–axis dimension was less than or equal to 384, phoneme recognition performance of the method proposed in this study degraded compared to that of AALBERT for both the linear and single hidden layer classifier classifiers. In contrast to the tendency observed in the phoneme recognition results, the accuracy of the speaker recognition task decreased with an increase in the time–axis dimension. In addition, the accuracy of the up-sampled case was lower than that of the down-sampled case.

This may be attributed to the following reasons: First, as the speaker information does not change within a single sentence, the model focused on the speaker information when long sequences were down-sampled into shorter representations. Second, as phonetic information changed faster than speaker information, the up-sampling process enhanced the encoding of those changes by the model. There was no significant difference between the accuracies of the "up-sampled" and "down-sampled" cases of the phoneme recognition task. In contrast, the accuracy for the "down-sampled" case of the speaker recognition task was significantly higher than that of the "up-sampled" case, indicating that the down-sampling process facilitated the modelling of the speaker information during the pre-training.

TABLE 4. Performance of downstream tasks with a change in the time-axis dimension (frame-level speaker recognition, phoneme recognition with one hidden layer classifier, and linear classifier).

		Time-axis dimension				
		192	384	768	1536	AALBERT
Speaker Accuracy (%) Frame	Total	**99.07**	97.44	96.99	92.86	93.21
	Up	86.66	86.60	91.65	**92.88**	-
	Down	**99.10**	97.53	97.28	92.82	-
Phoneme Accuracy (%) 1Hidden	Total	58.23	59.22	61.63	**64.25**	60.97
	Up	56.48	59.75	61.82	**64.31**	-
	Down	58.32	59.19	61.48	**63.14**	-
Phoneme Accuracy (%) Linear	Total	49.89	50.58	51.78	**52.59**	50.40
	Up	49.76	51.42	52.19	**52.20**	-
	Down	49.90	50.53	51.65	**52.60**	-

TABLE 5. Comparison of the speaker recognition accuracy (train-clean-100h).

	Mel-80	MockingJay	AALBERT	Proposed
Sentence (%)	81.09	**99.82**	99.71	**99.82**
Frame (%)	9.14	92.51	93.21	**99.07**
# of speakers	251			

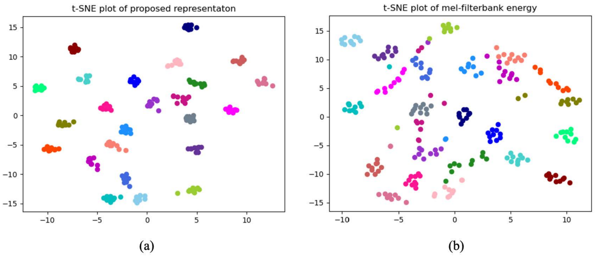


FIGURE 5. t-Distributed stochastic neighbor embedding plots of 24 randomly selected speakers in the *train-clean-100h* dataset: (a) proposed representation, (b) mel-filterbank energies.

TABLE 6. Comparison of the frame-wise phoneme accuracies of the various models.

	Mel-80	MockingJay	AALBERT	Proposed
Linear (%)	37.34	**52.84**	50.40	52.59
1Hidden (%)	41.43	62.57	60.97	**64.25**

b: SPEAKER CLASSIFICATION

A speaker recognition task was performed using conventional speech features and deep learning-based speech representations. For this, four types of features were compared: conventional 80 dimensional mel-filterbank energies, representations learned from AALBERT, representations learned from MockingJay, and representation learned from the proposed LN representation model. The time-axis dimension of the LN representation model is set to 1536. Various experiments were conducted on sentence and frame level predictions using the 100 h clean subset of LibriSpeech (i.e., *train-clean-100h*) by dividing the subset into train, development, and test sets at a ratio of 8:1:1. For the sentence-level predictions, the obtained frame-based representations were averaged, after which the averaged feature was delivered into the speaker classification network. For the frame-level prediction, a speaker ID was predicted in each frame using the represented feature.

Table 5 shows the speaker recognition accuracy of each of the methods for the sentence- and frame-level cases. The time-axis dimension for the proposed model is set to 1536 which showed best performance in previous experiment. For the sentence case, all the systems exhibited high performance. In addition, the representations of the frame-level case from our proposed model achieved an accuracy of 99.07%, which was higher than those from the baseline methods and mel-filterbank features in the *train-clean-100h* dataset. This could be attributed to the fact that the re-sampling operation enabled the LN representation learning model proposed in this study to handle time-scaled information with various ratios. Consequently, the learned features exhibited better generalization properties in the speaker recognition task.

Fig. 5 shows the t-Distributed Stochastic Neighbor Embedding plot of 24 randomly selected speakers from the *train-clean-100h* dataset. Ten sentences were randomly selected for each speaker, and their latent representations and mel-filterbank features were averaged in the time domain, which is identical to the process used to obtain the input to the sentence-level speaker recognition task. The results revealed that the latent representations extracted from the proposed model exhibited better speaker separation compared to the mel-filterbank energy features. The latent representations from the same speaker ID from the proposed model were more closely clustered, which corresponded to the higher accuracy of the model on the speaker recognition task.

c: PHONEME CLASSIFICATION

The performance of representations from the proposed model on a phoneme classification task was evaluated. For this, aligned phoneme labels and the train/test split setting from CPC [16] were utilized in the experiments. As the re-sampling operation changed the duration of each segment and resulted in an asynchronization between the frame-wise labeled phoneme IDs, this process was excluded during testing. The time–axis dimension of the proposed model was set to 1536, which provided the best performance in the phoneme recognition task, and the results are summarized in Table 6. Compared to AALBERT, the proposed model exhibited an enhanced performance, which could be attributed to its data augmentation effect. In addition, the model proposed in this study exhibited comparable performance to Mockingjay, which utilized a larger number of parameters, on a linear classifier, and an enhanced performance on the classifier with a single hidden layer.

C. APPLICATIONS: ASR & TTS

In this section, the performance of the unified ASR and TTS system introduced in Section III-B was discussed. The unified ASR and TTS system converted speech spectra into phonetic information in the ASR module and phonetic information into speech spectra in the TTS module in

TABLE 7. Network architecture of discriminators and generators.

	Discriminators	Generators	
Block 0	Conv2d(1, 32, 3, (1,3), 1)	Down-sampling rate	1
Block 1	Conv2d(32, 64, 7, 4, 2)	# of Layers	3
	BatchNorm2d(64)	# of attention heads	12
	LeakyReLU(0.2)	Hidden size	768
Block 2	Conv2d(64, 128, 4, 2, 1)	FF size	3072
	BatchNorm2d(128)	Shared layers	True
	LeakyReLU(0.2)	Time-axis dimension	768
Block 3	Conv2d(128, 256, 4, 2, 1)	Speech representation	(256, 768)
	BatchNorm2d(256)	Text representation	(256, 768)
	LeakyReLU(0.2)		
Block 4	Conv2d(256, 512, 4, 2, 1)		
	BatchNorm2d(512)		
	LeakyReLU(0.2)		
Block 5	Conv2d(512, 128, 4, 2, 1)		
	BatchNorm2d(128)		
	LeakyReLU(0.2)		
	Conv2d(128, 1, 4, 1, 0)		

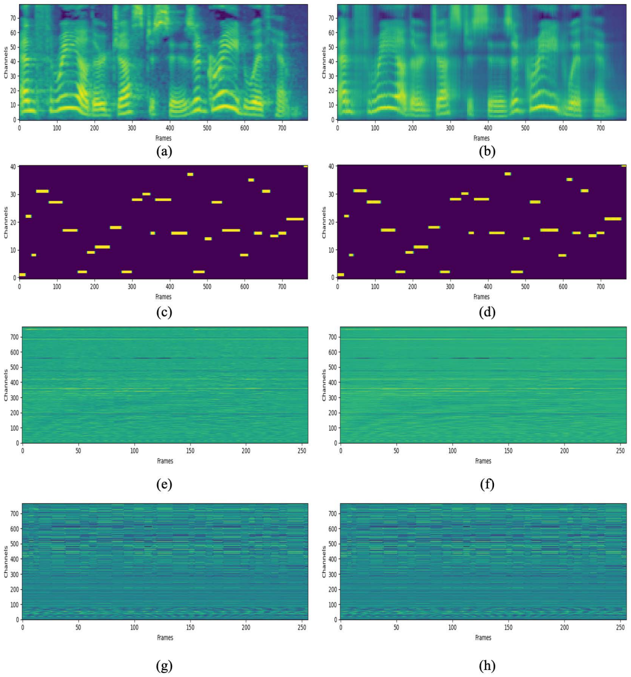
the LN representation domain. To demonstrate the feasibility of the proposed transformation model on a single speaker, experiments were conducted on the LJSpeech database [34], which consists of a 24 h speech recorded by a female English speaker.

For the unified system, first, LN text representations were obtained using the proposed model. Subsequently, the input features of the text representations were obtained using a series of one-hot vectors. In addition, phoneme transcriptions predicted by the Montreal Forced Aligner were utilized [26]. The text representation model was trained using the same structure used for the speech representations. The length distribution of the LJSpeech consisted of 13100 sentences, wherein the minimum, maximum and mean length of the sentences were 1.11, 10.10, and 6.58 sec, respectively. Considering the distribution of the sentence lengths and the change in the performance with a change in the length, which was discussed in the previous section, the time–axis dimension of the representation was set to 768. Examples of the reconstructed text features are shown in Appendix.

1) NEURAL VOCODER & MODEL SETTING

For the TTS module, we utilized the pre-trained HiFi-GAN vocoder [31] to synthesize speech waveforms using speech spectra as an auxiliary parameter. In addition, a 64 ms analysis window and 16 ms frame shift were utilized to estimate the spectra, and the minimum and maximum frequencies of the mel-filterbanks was set to 0 and 8000 Hz, respectively. The architectures of the discriminators and generators of the CycleGAN model are summarized in Table 7. The parameters of the convolution layer described in the table are the number of channels for the input, output, kernel size, stride, and padding size. The ASR and TTS generators consisted of three transformer layers that share parameters across the layers.

Because the decoded output of the reference representation was used for the loss, the parameters of the representation models were fixed while training the transformation network. Furthermore, the Adam optimizer with a learning rate of 0.0002 was utilized and trained with a batch size of 8. In addition, to enhance the performance of the model, a scheduled learning was applied to λ_{pred}. The λ_{pred} value was increased from 0.1 to 0.5, then to 0.9, and finally to 0.99 at 10, 50, and 150 k steps, respectively. Furthermore, the setup was empirically determined based on various experiments when the prediction and decoder losses remained constant; however, there was an increase in the total losses owing to the loss term of the generator.



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 6. Examples of (a) Decoded mel-spectrogram from the reference speech representation, (b) decoded mel-spectrogram from the predicted speech representation, (c) decoded text feature from the reference text representation, (d) decoded text feature from the predicted text representation, (e) reference speech representation, (f) predicted speech representation, (g) Reference text representation, (h) predicted text representation. The conversion occurs between the representations domain, the from the reference text representation, (g), to speech representation, (f), in TTS and from the reference speech representation, (e), to text representation, (h), in ASR.

Examples of the representations and decoded speech and text outputs in the unified ASR–TTS system are shown in Fig. 6. In the ASR prediction of the system, the decoded text features from the reference and predicted text representations (Fig. 6 (c) and (d)) were almost identical. In addition, as discussed in the next section, this significantly increases the accuracy of the phoneme recognition in the ASR task. However, the comparison of the decoded mel-spectrograms from the reference and the predicted speech representations of the TTS prediction (Fig. 6 (a) and (b)) revealed the presence of two major artifacts in the predicted sample. One of the artifacts was a smoothed spectrum in high-frequency regions, and the other was an unstable F0 trajectory, particularly in the low F0 regions, which originated from the prediction error in the representation domain. These problems can be addressed by training the entire TTS system (i.e., both the

TABLE 8. Results of the ASR-TTS application ('extracted' uses extracted speech or text features, 'Reconstructed' uses reconstructed mel-spectrograms or text feature using the proposed length normalized model, 'Predicted' uses decoded output from the predicted transformed speech or text representations).

		F0RMSE (Hz)	V/UV Error(%)	LSD (dB)	PER (%)
TTS	Extracted	5.49	6.08	-	
	Reconstructed	5.60	6.11	0.92	
	Predicted	28.41	17.02	3.40	
ASR	Reconstructed	-			1.84
	Predicted				7.15

representation learning model and the neural vocoder) in an end-to-end manner, or by adding more network layers to the text-to-spectrum representation prediction process. However, these methods were not included in this study.

2) PERFORMANCE OF THE UNIFIED ASR-TTS SYSTEM

The performance of the unified ASR and TTS model on the LJSpeech database is discussed in this section. The performance of the model for ASR task was evaluated by measuring the frame-wise phoneme error rate (PER). The CMU-based phoneme set [35] was used for training and testing. The performance of the of the model for TTS task was evaluated using log-spectral distance (LSD), F0 root mean-square error (F0RMSE), and V/UV classification error rate. LSD measured the distance between the original and predicted mel-spectrogram in a log-scale; F0RMSE measured the distance between the fundamental frequencies of the original and predicted speech; and V/UV error measured the classification accuracy of voice/unvoiced flags in each frame. The voiced/unvoiced error was measured from the V/UV flags obtained from the clean and generated speech, whereas the F0RMSE and LSD were measured using only voiced speech segments.

LSD is measured as follows:

$$D_{LS} = \sqrt{\frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\left[10\log_{10}\frac{P(t,k)}{\hat{P}(t,k)}\right]^2}, \qquad (15)$$

where $P(t,k)$, and $\hat{P}(t,k)$ stand for the ground truth and generated power spectrum, and t and k represent the time and frequency bin index respectively.

F0RMSE is a Euclidean distance between the fundamental frequencies of reference and synthesized speech, and it can be obtained as follows:

$$F0RMSE[Hz] = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(F0_x(t) - F0_{\hat{x}}(t))^2}, \qquad (16)$$

where $F0_x$ and $F0_{\hat{x}}$ denote the original and the estimated fundamental frequencies, respectively.

The results of the experiments are summarized in Table 8. The ASR module of the unified system achieved a PER of 1.84 and 7.15% on the reconstructed and predicted cases in the test set, respectively, which corresponded to a high phoneme recognition accuracy of 98.16 and 92.85% respectively. In addition, the F0RMSE and V/UV error rate of the TTS module were 5.49 Hz and 6.08%, respectively, when the ground truth mel-spectrograms were used for the pre-trained HiFi-GAN vocoder. The reconstructed mel-spectrogram input case that utilized the decoded output of the proposed representation model as a conditional input for the vocoder exhibited comparable performance to the reference case. In addition, owing to the significantly low reconstruction error of the proposed speech representation model, it had no significant effect on the synthesized speech quality. However, there was a degradation in the performance of the predicted mel-spectrogram case that utilized the decoded output of the predicted speech representation (i.e., 3.40 dB of LSD and increased F0RMSE and V/UV error rate), which could be attributed to the artifacts mentioned in the previous subsection.

To investigate the performance of the system in a multi-speaker environment, additional experiment was conducted on *LibriSpeech* dataset. The *train-clean-100h* database was used for the training and *test-clean* was used for testing. The performance of the multi-speaker TTS system was not measured because the speaker embedding layer is not included in proposed framework. The frame-wise PER of the recognition model for the reconstructed and predicted cases was 1.07 and 22.32%, respectively, indicating to a degradation in the performance of the model compared to the that of the LJSpeech case. However, owing to the fact that fewer training data were used in the system compared to typical ASR scenarios (960 h), and that a language model was not used, the performance of the system can be further enhanced.

Although the quality of the synthesized speech from the TTS module was not comparable to that of modern state-of-the-art systems, the results were expected owing to the experimental settings used in this study. In addition, as the parameters of HiFi-GAN were not updated jointly while training the ASR and TTS generators, we believe that there were significant mismatches between the characteristics of the generated mel-spectrograms and those of the ground truth auxiliary mel-spectrogram features that the neural vocoder was previously trained on. As previously stated, this performance degradation can be addressed by training the entire system, including the neural vocoder, in an end-to-end manner or by including additional network layers during the TTS prediction process. However, these measures were not employed in this study because this study was not aimed at achieving a state-of-the-art TTS performance. In contrast, this study aimed to demonstrate the practicability of the LN representation learning process proposed in the study on a unified S2S generation framework without using any recurrent processes.

V. CONCLUSION

In this study, we proposed a LN representation learning method that can encode arbitrary length sequences into fixed-shape representation vectors in a self-supervised manner. Unlike traditional attention-based encoder–decoder models, the proposed method can output representations for entire input sequences without using an auto-regressive

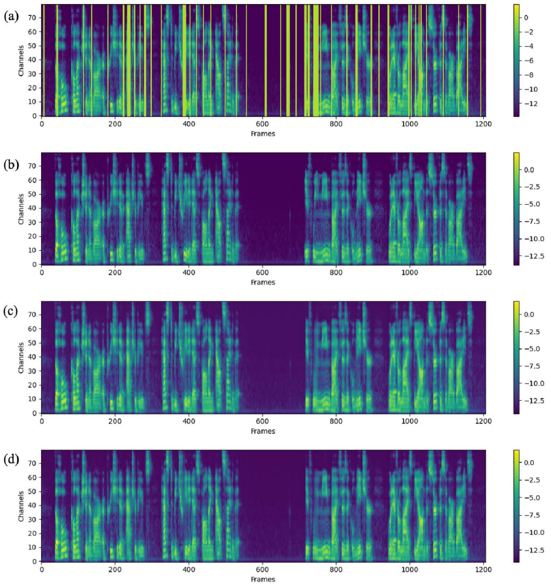


FIGURE 7. Examples of acoustic features: (a) Masked mel-filterbank energy, (b) ground truth mel-filterbank energy, (c) reconstructed mel-filterbank energy from the proposed length-normalized representation model, (d) reconstructed mel-filterbank energy from AALBERT.

process, which enabled the parallelization of the decoding process. The proposed method was applied to speech and text data, and it exhibited comparable performance to conventional transformer-based representation learning methods on a reconstruction task. In addition, the proposed representation achieved better performance on downstream tasks, such as speaker and phoneme recognition, which could be attributed to the effect of the augmentation effect from the training process on the time-scale of the input sequences. The representation vectors were applied to build a unified ASR and TTS system, which was trained using a CycleGAN style framework, which demonstrated a strong performance on the ASR part of the system and exhibited potential to generate high-quality synthesized speech on the TTS part.

APPENDIX

EXAMPLES OF THE RECONSTRUCTION TASKS

Fig. 7 shows the masked, ground truth, and predicted mel-spectrogram outputs from the proposed model and AALBERT. These spectrogram examples are provided to visually compare the degree of difference in the reconstruction ability of both models. As shown in Fig. 7 (a), which shows the input of the model, vertical masks that block the entire frequency band at specific times are randomly distributed. These masked regions are almost perfectly reconstructed from the remaining unmasked information in both proposed LN representation model and AALBERT model, as shown in the reconstructed spectrum. This indicates that there is no significant difference between Fig. 7 (b) and (c), which show the ground truth and predicted output of the proposed model, respectively. In addition, considering the dynamic range of the speech features, the comparison of Fig. 7 (c) and (d) reveal that there is no significant difference in the reconstruction loss of the two systems, which is listed in Table 3.

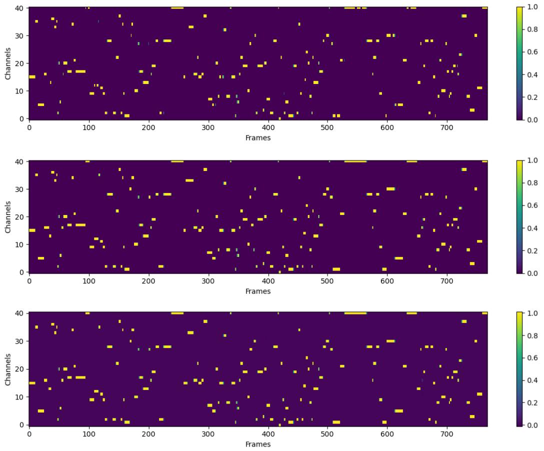


FIGURE 8. Examples of text features: (a) masked one-hot encoded, (b) ground truth, (c) reconstructed.

Fig. 8 shows examples of the input, ground truth, and reconstructed text features. The text feature was derived from the phoneme transcription, which was predicted from the Montreal Forced Aligner. Subsequently, the feature is converted into a one-hot vector depending on the phoneme ID, as shown in the Fig. 8 (b). The horizontal and vertical axes correspond to the frame index and phoneme class ID, respectively. When the time–axis dimension was 768, the reconstruction loss of the LN text representation learning model was 1.59e-3. Compared to the speech representation model, training was significantly harder in the proposed model owing to the discontinuous characteristics of the text features. Moreover, our preliminary experiments revealed that the details of the text features cannot be restored if the reconstruction loss is greater than 0.01. In these situations, only silence and vowels with long duration can be successfully reconstructed. In the final version of the text representation model, similarly to the results on speech representations, the proposed model enabled the accurate retrieval of the masked information in the reconstructed text features.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[3] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomed. Signal Process. Control*, vol. 59, May 2020, Art. no. 101894.

[4] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[5] R. Zhang, H. Wu, W. Li, D. Jiang, W. Zou, and X. Li, "Transformer based unsupervised pre-training for acoustic representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6933–6937.

[6] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2041–2053, Dec. 2019.

[7] A. H. Liu, T. Tu, H.-Y. Lee, and L.-S. Lee, "Towards unsupervised speech recognition and synthesis with quantized speech representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7259–7263.

[8] S. Parekh, S. Essid, A. Ozerov, N. Q. K. Duong, P. Perez, and G. Richard, "Weakly supervised representation learning for audio-visual scene analysis," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 416–428, 2020.

[9] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6429–6433.

[10] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," 2020, *arXiv:2006.10029*.

[11] S. Parthasarathy and C. Busso, "Semi-supervised speech emotion recognition with ladder networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2697–2709, 2020.

[12] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6707–6717.

[13] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[14] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*.

[15] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3497–3501.

[16] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[17] A. Baevski, S. Schneider, and M. Auli, "Vq-wav2vec: Self-supervised learning of discrete speech representations," 2019, *arXiv:1910.05453*.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[19] A. T. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6419–6423.

[20] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-Y. Lee, "Audio albert: A lite bert for self-supervised learning of audio representation," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 344–350.

[21] A. T. Liu, S.-W. Li, and H.-y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," 2020, *arXiv:2007.06028*.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[23] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6166–6170.

[24] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 976–989, 2020.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[26] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, Aug. 2017, pp. 498–502.

[27] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, and W.-Y. Ma, "Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6609–6618.

[28] R. Zheng, J. Chen, M. Ma, and L. Huang, "Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation," 2021, *arXiv:2102.05766*.

[29] W. Cao, Q. Lin, Z. He, and Z. He, "Hybrid representation learning for cross-modal retrieval," *Neurocomputing*, vol. 345, pp. 45–57, Jun. 2019.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[31] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," 2020, *arXiv:2010.05646*.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[34] K. Ito and L. Johnson. (2017). *The LJ Speech Dataset*. [Online]. Available: https://keithito.com/LJ-Speech-Dataset

[35] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. ISCA Workshop Speech Synth.*, 2004, pp. 1–2.

[36] G. Boulianne, "A study of inductive biases for unsupervised speech representation learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2781–2795, 2020.

[37] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.

[38] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey," *IEEE Access*, vol. 7, pp. 63373–63394, 2019.

[39] B. Wu, W. Chen, Y. Fan, Y. Zhang, J. Hou, J. Liu, and T. Zhang, "Tencent ML-images: A large-scale multi-label image database for visual representation learning," *IEEE Access*, vol. 7, pp. 172683–172693, 2019.

[40] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Span-BERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, Jan. 2020.

[41] J. Xu, X. Tan, Y. Ren, T. Qin, J. Li, S. Zhao, and T.-Y. Liu, "LRSpeech: Extremely low-resource speech synthesis and recognition," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2802–2812.

KYUNGGUEN BYUN received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2013, where he is currently pursuing the Ph.D. degree. He was an Intern at Microsoft Research Asia, Beijing, China, in 2017; and Qualcomm, San Diego, CA, USA, in 2020. He has been working as a Researcher with Qualcomm, since 2021. His research interests include text-to-speech, speech enhancement, and representation learning.

SEYUN UM received the B.S. degree in electronic engineering from Soongsil University, Seoul, South Korea, in 2017. She is currently pursuing the Ph.D. degree with Yonsei University. During her master's degree, she worked on a project for the Electronic and Telecommunications Research Institute (ETRI). Her research interests include emotional text-to-speech, image-to-speech, and style transfer learning.

HONG-GOO KANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Yonsei University, South Korea, in 1989, 1991, and 1995, respectively.

From 1996 to 2002, he was a Senior Technical Staff Member with AT&T Labs-Research, Florham Park, NJ, USA. In 2002, he joined the Department of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. He worked for Broadcom, Irvine, CA, USA; and Google, Mountain View, CA, USA, from 2008 to 2009 and 2015 to 2016, respectively, as a Visiting Scholar, where he participated in various projects on speech signal processing. His research interests include speech/audio signal processing, audio–visual signal processing, and machine learning. He was an Associate Editor of the IEEE Transactions on Audio, Speech, and Language Processing, from 2005 to 2008; and served numerous conferences and program committees.