# CROSS-DIALECT TEXT-TO-SPEECH IN PITCH-ACCENT LANGUAGE INCORPORATING MULTI-DIALECT PHONEME-LEVEL BERT

*Kazuki Yamauchi, Yuki Saito, Hiroshi Saruwatari*

The University of Tokyo, Japan

## ABSTRACT

We explore *cross-dialect text-to-speech (CD-TTS),* a task to synthesize learned speakers' voices in non-native dialects, especially in pitch-accent languages. CD-TTS is important for developing voice agents that naturally communicate with people across regions. We present a novel TTS model comprising three sub-modules to perform competitively at this task. We first train a backbone TTS model to synthesize dialect speech from a text conditioned on phoneme-level accent latent variables (ALVs) extracted from speech by a reference encoder. Then, we train an ALV predictor to predict ALVs tailored to a target dialect from input text leveraging our novel multi-dialect phoneme-level BERT. We conduct multi-dialect TTS experiments and evaluate the effectiveness of our model by comparing it with a baseline derived from conventional dialect TTS methods. The results show that our model improves the dialectal naturalness of synthetic speech in CD-TTS.

***Index Terms*—** text-to-speech, self-supervised learning, pitch-accent, accent latent variable

## 1. INTRODUCTION

Pitch-accent is a crucial prosodic attribute for natural speech communication in pitch-accent languages. In Japanese, one of pitch-accent languages, each mora has its corresponding high or low (H/L) pitch-accent to distinguish homophones. For instance, both 雨 (rain) and 飴 (candy) have the same pronunciation あめ (a-me), but their pitch-accents ("HL" and "LH") distinguish these words in Tokyo-dialect. Therefore, typical Japanese text-to-speech (TTS) [1] models take as input accent labels obtained using accent dictionaries (Fig. 1).

Dialects in pitch-accent languages each have a different pitch-accent rule. For instance, in Osaka-dialect, one of Japanese dialects, "LH" pitch-accent is used to pronounce 雨 (rain). Therefore, it is essential for dialect TTS in pitch-accent languages to reproduce the pitch-accent of synthetic speech tailored to each dialect to avoid miscommunication. However, building accent dictionaries to obtain the accent labels corresponding to texts for various dialects is very costly. Indeed, accent dictionaries are only available for Tokyo-dialect in Japanese. Therefore, current TTS systems find it challenging to adapt the pitch-accent of synthetic speech to different dialects, and this challenge is not well explored.

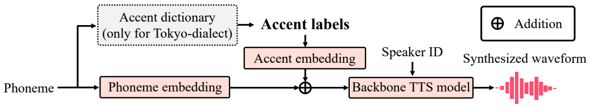

**Fig. 1**. Flowchart of typical Japanese TTS model.

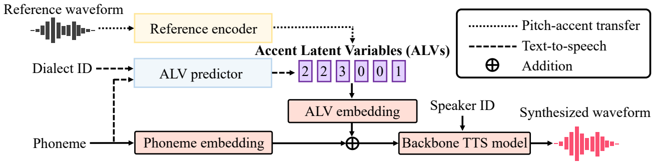

**Fig. 2**. Overview of our proposed TTS model.

In this paper, we explore a new task called *cross-dialect (CD)-TTS*, which aims to synthesize learned speakers' voices in a non-native dialect, especially in pitch-accent languages. CD-TTS is important for localizing TTS systems by adapting the pitch-accent of synthetic speech to regional dialects, leading to natural speech communication between computers and humans across regions. Note that CD-TTS differs from existing *cross-lingual TTS* [2]; specifically, CD-TTS focuses on several dialects within one specific language, which have similar but typically different pitch-accent systems and vocabularies. We propose a novel TTS model for CD-TTS as illustrated in Fig. 2, incorporating data-driven pitch-accent modeling using phoneme-level accent latent variables (ALVs). Our model can automatically predict ALVs tailored to each dialect instead of relying on accent dictionaries. Also, the ALV predictor incorporates a dialect-adapted version of phoneme-level BERT (PL-BERT) [3], multi-dialect (MD)-PL-BERT, to improve the accuracy of ALV prediction. The MD-PL-BERT is pre-trained on our constructed multi-dialect text corpus to capture both common and distinct textual features across dialects. We conduct Japanese multi-dialect TTS experiments and compare our model with a baseline derived from conventional dialect TTS methods. Audio samples are available on our demo page [1]. Our main contributions are as follows:

- We explore a new task denoted as CD-TTS to synthesize learned speakers' voices in a non-native dialect.

- We propose a novel TTS model for CD-TTS that automatically predicts ALVs tailored to each dialect from text, leveraging our novel MD-PL-BERT.

[1] https://kyamauchi1023.github.io/yamauchi24slt

- We present the result of evaluation experiments and demonstrate that leveraging our ALV predictor improves the dialectal naturalness of synthetic speech in CD-TTS.

## 2. RELATED WORK

### 2.1. Prosody transfer

Prosody transfer [4]–[7] is a technology to adapt prosody of synthetic speech to match that of reference speech while maintaining the speaker's voice timbre. Typical prosody transfer methods extract speaker-independent latent representation of prosody from speech by a variational autoencoder (VAE) [8]-based reference encoder. For example, Accent-VITS [9], a prosody transfer method for Chinese accented speech synthesis, extracts bottleneck (BN) features as prosody features from pre-trained automatic speech recognition (ASR) model and encodes them into latent representation by a VAE encoder.

### 2.2. Data-driven pitch-accent modeling

To address the challenge of Japanese dialect TTS, caused by the absence of accent dictionaries, Yufune et al. [10] proposed a TTS method utilizing ALVs, instead of accent labels. They first trained vector-quantized (VQ)-VAE [11] to extract mora-level quantized latent representation from prosody features such as fundamental frequency (F0) of speech. Since the representation can be regarded as pseudo accent label, they defined it as ALV. They showed that VQ-VAE was more efficient than VAE used in typical prosody transfer methods for accurately reproducing the natural pitch-accent of synthetic speech in Japanese. Then, they trained a TTS model conditioned on ALVs. Also, they trained an ALV predictor that takes an input text and predicts ALVs corresponding to each mora.

### 2.3. Self-supervised pre-training on text data for TTS

It has been demonstrated that leveraging self-supervised pre-training on text data, such as PnG BERT [12] and PL-BERT [3], effectively improves the prosodic naturalness of synthetic speech by TTS. PnG BERT is pre-trained on text data in a self-supervised manner, taking phonemes and graphemes of text as input. PL-BERT, on the other hand, does not take graphemes as input; instead, it is pre-trained to predict graphemes from phonemes, aiming to enhance the robustness of prosody prediction for unknown graphemes not present in the training data. In the context of Japanese Tokyo-dialect TTS, Japanese PnG BERT [13] improves the naturalness of pitch-accent of synthetic speech by pre-training to predict accent labels obtained using accent dictionaries.

### 2.4. Problems of conventional methods for dialect TTS

Yufune et al.'s study [10] focused on single-speaker intra-dialect TTS (ID-TTS), i.e., synthesizing speech in the same dialect as the target speaker's native dialect. Indeed, their model does not contain the functions to predict pitch-accent tailored to different dialects or adapt pitch-accent of synthetic speech to match that of an arbitrary speaker's reference speech. Also, while they demonstrated that the naturalness of speech synthesized using ALVs extracted from ground-truth speech has improved, the naturalness of speech synthesized using predicted ALVs was lower than that of speech synthesized without ALVs, due to the low accuracy of ALV prediction. Note that it has been demonstrated that inaccurate accent labels generally degrade the naturalness of synthetic speech [14]. One possible reason for the low ALV prediction accuracy of their model is the limited size of existing Japanese dialect speech corpora (e.g., CPJD [15]), which restricts the available data for training. However, constructing speech corpora with a sufficient amount of data for each dialect is very costly. Therefore, a method to improve the accuracy of ALV prediction without relying on additional dialect speech corpora is demanded.

Self-supervised pre-training on text data can be expected to improve the naturalness of pitch-accent for dialect TTS. However, current text pre-training methods for TTS [13] typically utilize texts written in the standard language as training data, lacking mechanisms to learn features that vary across dialects. Moreover, the availability of text corpora annotated with the dialect ID remains limited in size. Therefore, a self-supervised pre-training method that is effective for dialect pitch-accent prediction is demanded for multi-dialect TTS.

## 3. METHOD

Fig. 3 illustrates the architecture of the proposed dialect TTS model, comprising: 1) a backbone TTS model, 2) a reference encoder, and 3) an ALV predictor. The backbone TTS model synthesizes dialect speech conditioned on ALVs obtained by either of the other two modules. The reference encoder extracts ALVs, phoneme-level quantized latent representation of prosody. The ALV predictor predicts ALVs corresponding to each phoneme conditioned on a dialect ID. The ALV predictor incorporates our novel MD-PL-BERT, pre-trained on our constructed multi-dialect text corpus, to capture both common and distinct textual features across dialects and to predict pitch-accent for phrases unique to each dialect. Our model can synthesize speech from input text and a dialect ID by automatically predicting ALVs tailored to the target dialect (i.e., TTS). Additionally, by inputting an arbitrary speaker's reference speech with the desired pitch-accent, the pitch-accent of the synthetic speech can be adapted to match that of the reference speech (i.e., pitch-accent transfer).

### 3.1. Reference encoder

The reference encoder is a module for extracting ALVs from prosody features of reference speech, enabling data-driven pitch-accent modeling without reliance on accent dictionaries. We employ a VQ-VAE-based reference encoder, following Yufune et al.'s study [10]. Note that while Yufune et al. defined ALV at the mora-level [10], we define it at the phoneme-level.

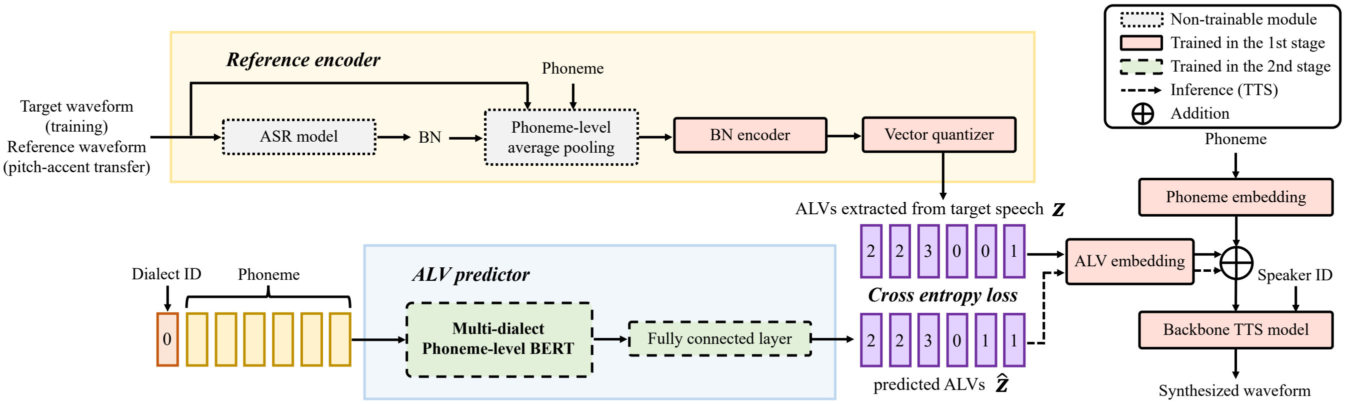

**Fig. 3**. The architecture of our proposed model, consisting of a reference encoder and an ALV predictor. In the first training stage, the reference encoder and backbone TTS model are trained. In the second training stage, the ALV predictor is trained.

To obtain prosody features related to pitch-accent information, the reference encoder incorporates a pre-trained ASR model into the ALV extraction framework, similar to the approach used in Accent-VITS [9]. Because pitch-accent is necessary for distinguishing words in pitch-accent languages, features obtained from a pre-trained ASR model are expected to contain sufficient prosody information.

Specifically, we first feed reference speech into the ASR model to extract BN features as the output of the ASR model's encoder's final layer. BN features are aggregated into phoneme-level features using average pooling, guided by phoneme alignment information, to obtain phoneme-level ALVs. Subsequently, they are fed into a one-dimensional convolutional neural network (1D-CNN)-based BN encoder. Finally, the encoder outputs are quantized by a VQ module to obtain the quantized indices, i.e., the ALVs.

### 3.2. ALV predictor incorporating MD-PL-BERT

The ALV predictor is a module to predict ALVs tailored to a target dialect from input text. We focus on PL-BERT [3], a self-supervised learning model pre-trained on text data, to improve the accuracy of ALV prediction. However, the original PL-BERT lacks mechanisms for learning linguistic features that vary across different dialects, making it challenging to predict ALVs specific to each dialect. To address this, we propose MD-PL-BERT, a dialect-adapted version of PL-BERT, and incorporate it into the ALV predictor. The pre-training strategy is similar to PL-BERT, but with two key differences.

First, we introduce conditioning PL-BERT on dialect ID, an identifier that indicates which dialect the input text is written in. Specifically, we add a dialect ID to the beginning of the input phoneme sequence to enable PL-BERT to learn linguistic features tailored to the specified dialect.

Second, we construct a large-scale multi-dialect text corpus and pre-train MD-PL-BERT on them. While pre-training MD-PL-BERT requires large-scale multi-dialect text corpora, the available text corpora annotated with dialect ID are limited in size. Recent research has demonstrated the effectiveness of using large language models (LLMs) for dialect translation and has proposed a method for the automated construction of dialect text corpora [16]. Inspired by this approach, we construct a multi-dialect text corpus by leveraging the data augmentation through translating texts written in the standard language (i.e., Tokyo-dialect in Japanese) into a target dialect using an LLM. Specifically, we prompt a pre-trained LLM to translate a given Tokyo-dialect sentence into the target dialect using the following prompt: *"Rewrite the following sentences as if they were in [target dialect]: [sentence written in Tokyo-dialect]"*. The ALV predictor comprises MD-PL-BERT, pre-trained on this corpus, followed by a fully connected layer that predicts the ALVs from the output of the final layer of MD-PL-BERT.

### 3.3. Training and inference

Our model is trained in two stages. In the first stage, the reference encoder and the backbone TTS model are jointly trained while the parameters of the pre-trained ASR model remain frozen. The loss function is the sum of the losses from the backbone TTS model and the VQ loss [11]. During training, the target ground-truth speech is used as the reference speech. In the second stage, the ALV predictor is initialized with the pre-trained MD-PL-BERT and fine-tuned together with a fully connected layer. The loss function $\mathcal{L}$ used to train the ALV predictor is the cross-entropy loss (CELoss) between the ALVs extracted from the target speech by the reference encoder, $z$, and the predicted ALVs, $\hat{z}$, denoted as:

$$\mathcal{L} = \mathrm{CELoss}(z, \hat{z}) \tag{1}$$

During inference, our TTS model enables pitch-accent transfer by synthesizing speech using ALVs extracted from an arbitrary speaker's reference speech. This allows for control of the pitch-accent of synthetic speech by inputting reference speech with the desired pitch-accent. Pitch-accent transfer can be seen as a variant of prosody transfer. The key difference is that prosody transfer primarily focuses on emotion or speaking style, whereas pitch-accent transfer targets pitch-accent, which is discrete and more akin to linguistic information.

## 4. EXPERIMENTS

We evaluate our method in both ID-TTS and CD-TTS. The experiments focus on synthesizing speech in Osaka-dialect, one of Japanese dialects, by a native Osaka-dialect speaker (i.e., ID-TTS) and a Tokyo-dialect speaker (i.e., CD-TTS).

### 4.1. Experimental conditions

**Training dataset:** We used JSUT [17] and JMD [2] [18]. JSUT consists of approximately 7,700 utterances by a single Tokyo-dialect speaker (female), while JMD includes 1,300 utterances by native dialect speakers for each dialect. We mixed JSUT and the JMD-Osaka subset including voices by a single native Osaka-dialect speaker (female) and divided this mixed dataset into training (8,484 utterances), validation (256 utterances), and test (256 utterances) subsets.

**Evaluation dataset:** To evaluate the effectiveness of pitch-accent transfer using reference speech by an unseen speaker not present in the training dataset, we used speech in CPJD [15] as reference speech. CPJD is a multi-dialect speech corpus collected through crowdsourcing, containing 250 utterances for each dialect. We used the CPJD-Osaka subset including voices by a single native Osaka-dialect speaker (male) as reference speech for pitch-accent transfer.

**Training setup:** BN features were extracted by the encoder of the pre-trained Whisper large-v2 model[3] [19]. The phoneme alignment information to aggregate BN features into phoneme-level features was obtained using Julius [20]. The BN encoder first projects BN features aggregated at the phoneme level into 256 dimensions and feeds them into a stack of two 1D-CNN layers with a kernel size of 3, stride of 1, and filter size of 256. This process outputs phoneme-level 256-dimensional continuous vectors. Subsequently, the vectors are quantized into four classes, following the previous Japanese dialect TTS study [10]. Finally, the quantized vectors (i.e., ALV embeddings) are added to 256-dimensional phoneme embeddings. Note that the indices of the quantized vectors are the ALVs. The weight of the commitment loss in VQ loss [11] was set to 4.0. Also, we used FastSpeech 2 [21] as the backbone TTS model following the publicly available implementation (FastSpeech2-JSUT[4]) for the network architecture and training settings. That is, for the first stage of training, the model was trained with a batch size of 32, learning rate of 0.0625, and 100k iterations in 5 hours. The pre-trained HiFi-GAN UNIVERSAL_V1 model[5] [22] was used as a vocoder.

**Pre-training:** For pre-training MD-PL-BERT, we used Japanese Wikipedia corpus[6], containing approximately 1.0M documents, and ReazonSpeech small[7], containing approximately 62K utterances designed for building a Japanese ASR model. We used transcriptions in ReazonSpeech as text dataset written in Tokyo-dialect and translate them into Osaka-dialect. MD-PL-BERT was initialized by PL-BERT pre-trained on Wikipedia corpus and then pre-trained on transcriptions in ReazonSpeech with the data augmentation described in Section 3.2. We used Japanese Llama 2 [23][8], a.k.a., Swallow 13B[9] as the LLM for dialect translation. We followed the network architecture and pre-training strategy of PL-BERT described in the official implementation (PL-BERT[10]). To tokenize Japanese text into subwords, we used a publicly available tokenizer[11]. For grapheme-to-phoneme (G2P) conversion, we used OpenJTalk[12]. PL-BERT was pre-trained on Wikipedia corpus with a batch size of 8, learning rate of $4.0 \times 10^{-6}$, and 10M iterations in 10 days. MD-PL-BERT was pre-trained with a batch size of 16, learning rate of $5.0 \times 10^{-5}$, and 100k iterations in 10 hours. For the second stage of training the proposed model, it was trained with a batch size of 32, learning rate of 0.001, and 10k iterations in 5 hours.

**Model parameters and computational resources:** The backbone TTS model, the reference encoder, and the ALV predictor contained 35M, 790K, and 6M trainable parameters, respectively. All the models were trained on a single Nvidia A100 GPU using the Adam optimizer [24] with the linear scheduler of learning rate with warm up steps of 4000.

**Task definition and compared models:** We evaluated our proposed model through two tasks: 1) ID-TTS and 2) CD-TTS. The former and latter aim to synthesize speech 1) in the same dialect as the target speaker's native dialect and 2) in a different dialect from the target speaker's native dialect, respectively. The target speakers for ID-TTS and CD-TTS were defined as the JMD-Osaka speaker and the JSUT speaker, respectively. Input texts for TTS are sampled from transcriptions in CPJD-Osaka. We mainly evaluated the following models:

- **FS2 (baseline)**: The original FastSpeech 2
- **FS2-AP (proposed)**: The proposed model using ALVs predicted by the ALV predictor form input text
- **FS2-REF (proposed)**: The proposed model using ALVs extracted from reference speech

### 4.2. Evaluations

We conducted subjective and objective evaluations to compare the proposed model with an existing baseline.

**Mean opinion score (MOS) tests:** We conducted MOS tests via crowdsourcing to assess the naturalness of speech and the dialectal naturalness (i.e., *dialectality*) of pitch-accent for each method. Participants evaluated randomly selected synthetic speech samples by each method or natural speech samples in CPJD from two viewpoints: 1) naturalness (N-MOS)

[2] https://sites.google.com/site/shinnosuketakamichi/research-topics/jmd_corpus?authuser=0

[3] https://huggingface.co/openai/whisper-large-v2

[4] https://github.com/Wataru-Nakata/FastSpeech2-JSUT

[5] https://github.com/jik876/hifi-gan

[6] https://dumps.wikimedia.org/

[7] https://huggingface.co/datasets/reazon-research/reazonspeech

[8] https://llama.meta.com/

[9] https://huggingface.co/tokyotech-llm/Swallow-13b-instruct-hf

[10] https://github.com/yl4579/PL-BERT

[11] https://huggingface.co/tohoku-nlp/bert-base-japanese-whole-word-masking

[12] https://open-jtalk.sp.nitech.ac.jp

**Table 1**. Results of comparing the performance of FS2-AP-Scratch and FS2 in ID-TTS.

| A vs. B | Naturalness | Dialectality |
|---|---|---|
| FS2-AP-Scratch vs. FS2 | 0.250 vs. **0.750** | 0.227 vs. **0.773** |

and 2) dialectality (D-MOS). The former and latter mean whether 1) it sounds naturally human-like and 2) its pitch-accent sounds natural as Osaka-dialect, not Tokyo-dialect, on a 5-point scale from 1 (very unnatural) to 5 (very natural), respectively. For both ID-TTS and CD-TTS, 35 native Japanese speakers evaluated 24 randomly presented speech samples.

**Pairwise comparisons:** We also conducted several preference AB tests on the naturalness and dialectality of synthetic speech to determine the appropriate baseline and for ablation studies of the proposed model. Twenty listeners participated in the tests via crowdsourcing, and each listener evaluated ten pairs of synthetic speech samples. In the following subsection, **bold** values in the tables showing the results of the AB test indicate that significant differences are determined by a Student's $t$-test at a 5% significance level.

**Speaker similarity:** To verify that ALVs are speaker-independent, we measured the speaker similarity of synthetic speech to the target speaker's natural speech, using cosine similarity between x-vectors [25] (SIM). Specifically, we computed the mean of SIM between the averaged x-vector among all speech samples of the target speaker in the test set and the x-vector of each synthetic speech. We obtained x-vectors using a pre-trained model[13].

### 4.3. Results and discussion

**What is the appropriate baseline in this study?** As well as FS2, one can regard FS2-AP without initialing the model parameter on the basis of the MD-PL-BERT pre-training (i.e., FS2-AP-Scratch) as the candidate baseline. The reason is that the model structure and two-stage training of FS2-AP-Scratch are similar to those used in Yufune et al.'s study [10]. Therefore, we first compared these two methods in a preference AB test in ID-TTS. As shown in Table 1, FS2 significantly outperformed FS2-AP-Scratch, indicating that the prediction performance of ALV predictor without pre-training on text datasets is poor and inaccurate ALV prediction makes the naturalness of synthetic speech even worse. This result is consistent with the result of Yufune et al.'s study [10] mentioned in Section 2.4. From this result, we decided to use FS2 as the baseline to be compared with the proposed model.

**Can our models improve dialect TTS performance?** Table 2 shows the results of MOS tests. First, from the results of ID-TTS shown in Table 2(a), no significant difference in MOS was observed between FS2 and FS2-AP. Meanwhile, pitch-accent transfer through reference speech input tended to improve D-MOS, although the improvement was not statistically significant. Second, from the results of CD-TTS shown in Table 2(b), FS2-AP achieved significantly higher D-MOS than FS2. This indicates that the ALV predictor learned typical accent representation of Osaka-dialect, and the proposed model was effective in improving the dialectality of synthetic speech in CD-TTS. Furthermore, pitch-accent transfer through reference speech input (i.e., FS2-REF) significantly improved D-MOS compared to FS2. Also, using ALVs extracted from reference speech by a different speaker from the target speaker did not degrade the speaker similarity to the target speaker. This demonstrates that our model enables pitch-accent transfer through an unseen speaker's reference speech input.

**Table 2**. Results of MOS test with 95% confidence interval and computed SIM. REF represents the reference speech. **Bold** values are significantly higher than those of FS2 according to the results of a student's $t$-test at a 5% significance level.

(a) ID-TTS: Synthesis of Osaka-dialect speech by Osaka-dialect speaker

| Method | Target speaker | N-MOS (↑) | D-MOS (↑) | SIM (↑) |
|---|---|---|---|---|
| FS2 | JMD (Osaka) | 3.30 ± 0.12 | 3.22 ± 0.13 | 0.990 |
| FS2-AP | JMD (Osaka) | 3.31 ± 0.13 | 3.26 ± 0.13 | 0.991 |
| FS2-REF | JMD (Osaka) | 3.23 ± 0.12 | 3.30 ± 0.12 | 0.992 |
| REF | CPJD (Osaka) | **3.89 ± 0.14** | **4.38 ± 0.09** | - |

(b) CD-TTS: Synthesis of Osaka-dialect speech by Tokyo-dialect speaker

| Method | Target speaker | N-MOS (↑) | D-MOS (↑) | SIM (↑) |
|---|---|---|---|---|
| FS2 | JSUT (Tokyo) | 3.57 ± 0.13 | 2.62 ± 0.13 | 0.990 |
| FS2-AP | JSUT (Tokyo) | 3.52 ± 0.13 | **3.00 ± 0.15** | 0.990 |
| FS2-REF | JSUT (Tokyo) | 3.58 ± 0.12 | **3.05 ± 0.14** | 0.990 |
| REF | CPJD (Osaka) | **4.39 ± 0.10** | **4.32 ± 0.13** | - |

**Table 3**. Results of comparing the performance of FS2 and FS2-AP in CD-TTS by native Osaka-dialect speakers.

| A vs. B | Naturalness | Dialectality |
|---|---|---|
| FS2 vs. FS2-AP | 0.506 vs. 0.494 | 0.387 vs. **0.613** |

**Is the improvement significant for native Osaka-dialect speakers?** We asked eight native Osaka-dialect speakers to evaluate the naturalness and dialectality of synthetic speech by FS2 and FS2-AP in a preference AB test. As shown in Table 3, FS2-AP significantly outperformed FS2 in dialectality, while maintaining the naturalness. This result demonstrates the effectiveness of the proposed TTS model is perceivable for not only crowdsourced listeners but also native dialect speakers.

### 4.4. Ablation study

**Are BN features effective for pitch-accent transfer?** Instead of BN features, F0 can be used as a prosody feature for ALV extraction, similar to Yufune et al.'s method [10]. Therefore, we compared the two prosody features, BN and F0, in the preference AB tests. To obtain speaker-independent prosody features, we normalized F0 in an utterance-wise manner. In addition, we linearly interpolated unvoiced regions of F0 in the phoneme-level average pooling. We used WORLD [26] to extract F0 from speech. Also, we set the target speaker to

[13] https://github.com/sarulab-speech/xvector_jtubespeech

**Table 4**. Results of comparing the performance of pitch-accent transfer by FS2-REF using BN feature and F0 as prosody features.

| A vs. B | Naturalness | Dialectality |
|---|---|---|
| F0 vs. BN | 0.400 vs. **0.600** | 0.424 vs. **0.576** |

**Table 5**. BLUE@4 and BERTScore between Osaka-dialect sentences and original (Saitama-dialect) sentences or sentences translated into Osaka-dialect. **Bold** scores are better.

| Text | BLEU@4 (↑) | BERTScore (↑) |
|---|---|---|
| Original | 0.370 | 0.873 |
| Translated | **0.401** | **0.882** |

**Table 6**. Results of comparing the absence of data augmentation (DA) by LLM-based dialect translation in CD-TTS.

| A vs. B | Naturalness | Dialectality |
|---|---|---|
| w/o DA vs. w/ DA | 0.491 vs. 0.509 | 0.343 vs. **0.657** |

the JSUT speaker. The evaluation results are shown in Table 4. From this table, BN significantly outperformed F0 in both evaluation cases, demonstrating the effectiveness of BN for ALV extraction. One possible reason is that while F0 is an acoustic feature, BN features can be considered as linguistic features acquired through the ASR task.

**How do ALVs influence the pitch-accent of synthetic speech?** For TTS in pitch-accent languages, it is desirable that humans can easily correct errors in the pitch-accent of synthetic speech. Therefore, we analyzed how ALVs influence the pitch-accent of synthetic speech to investigate the controllability and interpretability of ALVs. Specifically, we extracted log F0 (logarithm of fundamental frequency) of synthetic speech and aggregated it at the phoneme-level. The distribution was then plotted for each corresponding ALV. We used FS2-REF with the target speaker being the JMD speaker and utilized the CPJD corpus as reference speech. The results are shown in Fig. 4. It can be observed that log F0 of synthetic speech varies according to ALV classes. Specifically, log F0 for the intervals corresponding to ALV value $0 < 1 < 2 < 3$ tends to increase in order. This suggests that ALVs can be interpreted as four categorical levels of pitch in synthetic speech and regarded as pseudo high-low pitch-accent labels.

**Is the data augmentation by LLM-based dialect translation effective in improving the dialectality score?** To verify the effectiveness of dialect translation by LLM as data augmentation, we initially conducted objective evaluations on translation accuracy. We utilized transcriptions from CPJD, which contains semantically parallel transcriptions in multiple dialects. Initially, we translated 250 transcriptions written in Saitama-dialect, the dialect closest to Tokyo-dialect within CPJD, into Osaka-dialect using an LLM. Subsequently, we measured the similarity between the translated transcriptions and those originally written in Osaka-dialect in CPJD using BLEU [27] and BERTScore [28]. As shown in Table 5, sentences translated by the LLM are more similar to Osaka-dialect than the original sentences. This result indicates that the LLM has the ability for dialect translation.

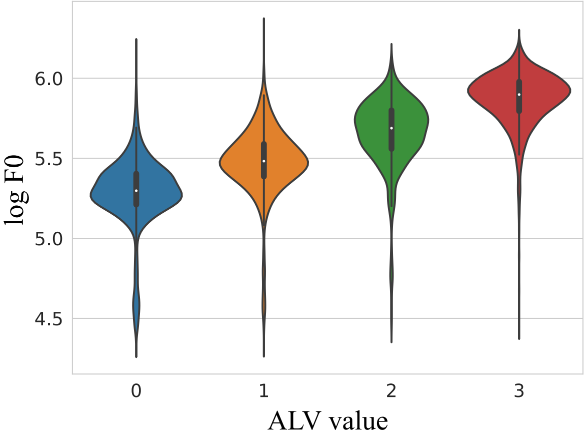

**Fig. 4**. The violinplot of logarithmic fundamental frequency (log F0) aggrigated by ALV value (0, 1, 2, or 3).

We also conducted a subjective evaluation to assess the effectiveness of our MD-PL-BERT compared to the original PL-BERT. Specifically, we compared two models in the preference AB tests: our FS2-AP incorporating MD-PL-BERT pre-trained with the data augmentation and the original PL-BERT pre-trained without the data augmentation. As shown in Table 6, our MD-PL-BERT, pre-trained on the multi-dialect text corpus constructed through the data augmentation, significantly improved the dialectality of synthetic speech.

## 5. CONCLUSIONS

We explored a new task called *cross-dialect text-to-speech (CD-TTS)*, which aims to synthesize learned speakers' voices in non-native dialects. To address this, we proposed a novel TTS model comprising three sub-modules designed to perform effectively in this task. We evaluated its performance not only on intra-dialect (ID)-TTS but also on CD-TTS through a series of subjective evaluations. The results show that our model improves the dialectality of synthetic dialect speech in CD-TTS without degrading the performance of ID-TTS.

In the future, we plan to investigate the effectiveness of our proposed model in dialect TTS using more dialects. We also plan to incorporate machine learning techniques used to enhance the performance of cross-lingual TTS, such as domain adaptation [29] and mutual information minimization [2], into our model for CD-TTS. Moreover, dialect TTS faces challenges not only with the lack of accent dictionaries but also with G2P converters. Data-driven modeling of phoneme labels without reliance on G2P converters is also a future task.

**Acknowledgements:** This work was supported by JST, ACT-X Grant Number JPMJAX23CB, Japan.

# References

[1] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. ICASSP*, New York, U.S.A., Apr. 1988, pp. 679–682.

[2] D. Xin, T. Komatsu, S. Takamichi, *et al.*, "Disentangled speaker and language representations using mutual information minimization and domain adaptation for cross-lingual TTS," in *Proc. ICASSP*, Montreal, Canada, Jun. 2021, pp. 6608–6612.

[3] Y. A. Li, C. Han, X. Jiang, *et al.*, "Phoneme-level BERT for enhanced prosody of text-to-speech with grapheme predictions," in *Proc. ICASSP*, Rhodes, Greece, Jun. 2023.

[4] R. Skerry-Ryan, E. Battenberg, Y. Xiao, *et al.*, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," in *Proc. ICML*, Jul. 2018, pp. 4693–4702.

[5] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *Proc. ICASSP*, Brighton, U.K., May 2019, pp. 5911–5915.

[6] V. Klimkov, S. Ronanki, J. Rohnke, *et al.*, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 4440–4444.

[7] S. Karlapati, P. Karanasou, M. Lajszczak, *et al.*, "CopyCat2: A single model for multi-speaker tts and many-to-many fine-grained prosody transfer," in *Proc. INTERSPEECH*, Incheon, South Korea, Sep. 2022, pp. 3363–3367.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, Banff, Canada, Apr. 2014.

[9] L. Ma, Y. Zhang, X. Zhu, *et al.*, "Accent-VITS: Accent transfer for end-to-end TTS," in *Proc. NCMMSC*, Suzhou, China, Dec. 2023.

[10] K. Yufune, T. Koriyama, S. Takamichi, *et al.*, "Accent modeling of low-resourced dialect in pitch accent language using variational autoencoder," in *Proc. SSW*, Budapest, Hungary, Aug. 2021, pp. 189–194.

[11] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. NIPS*, Long Beach, U.S.A., Dec. 2017, pp. 6309–6318.

[12] Y. Jia, H. Zen, J. Shen, *et al.*, "PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS," in *Proc. INTERSPEECH*, 2021, pp. 151–155.

[13] Y. Yasuda and T. Toda, "Investigation of Japanese PnG BERT language model in text-to-speech synthesis for pitch accent language," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1319–1328, 2022.

[14] K. Fujii, Y. Saito, and H. Saruwatari, "Adaptive end-to-end text-to-speech synthesis based on error correction feedback from humans," in *Proc. APSIPA ASC*, Chiang Mai, Thailand, Nov. 2022, pp. 1702–1707.

[15] S. Takamichi and H. Saruwatari, "CPJD Corpus: Crowdsourced parallel speech corpus of Japanese dialects," in *Proc. LREC*, Miyazaki, Japan, May 2018, pp. 434–437.

[16] A. A. A. Abdelaziz, A. H. Elneima, and K. Darwish, "LLM-based MT data creation: Dialectal to MSA translation shared task," in *Proc. OSACT Workshop*, Torino, Italia, May 2024, pp. 112–116.

[17] S. Takamichi, R. Sonobe, K. Mitsui, *et al.*, "JSUT and JVS: Free Japanese voice corpora for accelerating speech synthesis research," *Acoustical Science and Technology*, vol. 41, no. 5, pp. 761–768, Sep. 2020.

[18] S. Takamichi and H. Saruwatari, "JMD: Japanese multi-dialect corpus," 2021.

[19] A. Radford, J. W. Kim, T. Xu, *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, Hawaii, U.S.A., Jun. 2023, pp. 28 492–28 518.

[20] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proc. EUROSPEECH*, Aalborg, Denmark, Sep. 2001, pp. 1691–1694.

[21] Y. Ren, C. Hu, X. Tan, *et al.*, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, Vienna, Austria, May 2021.

[22] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, Virtual Conference, Dec. 2020, pp. 17 022–17 033.

[23] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, California, U.S.A., May 2015.

[25] D. Snyder, D. Garcia-Romero, G. Sell, *et al.*, "X-Vectors: Robust dnn embeddings for speaker recognition," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 5329–5333.

[26] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 7, pp. 1877–1884, Jul. 2016.

[27] K. Papineni, S. Roukos, T. Ward, *et al.*, "BLEU: A method for automatic evaluation of machine translation," in *Proc. ACL*, Philadelphia, U.S.A., Jun. 2002, pp. 311–318.

[28] T. Zhang, V. Kishore, F. Wu, *et al.*, “BERTScore: Evaluating text generation with BERT,” in *Proc. ICLR*, Virtual Conference, Apr. 2020.

[29] D. Xin, Y. Saito, S. Takamichi, *et al.*, “Cross-lingual text-to-speech synthesis via domain adaptation and perceptual similarity regression in speaker space,” in *Proc. INTERSPEECH*, Shanghai, China, Oct. 2020, pp. 2947–2951.