PROSODYBERT: SELF-SUPERVISED PROSODY REPRESENTATION FOR STYLE-CONTROLLABLE TTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose ProsodyBERT, a self-supervised approach to learning prosody representations from raw audio. Different from most previous works, which use information bottlenecks to disentangle prosody features from speech content and speaker information, we perform an offline clustering of speaker-normalized prosody-related features (energy, pitch, their dynamics, etc.) and use the cluster labels as targets for HuBERT-like masked unit prediction. A span boundary loss is also introduced to capture long-range prosodic information. We demonstrate the effectiveness of ProsodyBERT on a multi-speaker style-controllable text-to-speech (TTS) system. Experiments show that the TTS system trained with ProsodyBERT features can generate natural and expressive speech samples, surpassing the model supervised by energy and pitch on subjective human evaluation. Also, the style and expressiveness of synthesized audio can be controlled by manipulating the prosody features. In addition, We achieve new state-of-the-art results on the IEMOCAP emotion recognition task by combining our prosody features with HuBERT features, showing that ProsodyBERT is complementary to popular pretrained speech self-supervised models. [1]

1 INTRODUCTION

Human speech contains information beyond textual content. For example, the intonation, stress, rhythm, and tempo of speech carry important cues associated with the speaking style, emotion, and intent. These factors are generally referred as prosody. Prosodic modeling has been widely investigated in expressive text-to-speech (TTS) (Ren et al., 2020; Kenter et al., 2020; Ren et al., 2022) and voice conversion(VC) (Kreuk et al., 2021; Zhou et al., 2022), and has shown to be important for generating natural and expressive synthesized speech. Prosody is also applied in spoken language understanding tasks by providing complementary information that is not covered by text. Example tasks include parsing (Tran et al., 2017), disfluency detection (Zayats et al., 2019), and punctuation prediction (Cho et al., 2022).

The most commonly used prosody representations include acoustic attributes like fundamental frequency (F_0), energy, and duration. However, these features have several issues. First, pitch and energy are usually extracted by external signal-processing tools. The algorithms involve inevitable errors (e.g. inaccurate F_0 estimation) and are not robust in complex acoustic environments. Second, pitch, energy, and duration are interdependent on each other, and the relation may be potentially broken. Modeling them independently in downstream tasks may result in unnatural prosody. Learning-based prosody representations are proposed to address these issues. These methods generally rely on autoencoders that condition on text and speaker identity, which encourages residual information (assumed to be prosody) to be captured in an information bottleneck.(Skerry-Ryan et al., 2018; Wang et al., 2018; Zhang et al., 2019; Qian et al., 2020) However, these approaches rely on high-quality transcripts. Prosody has high variability across different styles and speakers, making it challenging to learn generalizable prosody representation from a limited amount of high-quality data.

Another paradigm of representation learning is self-supervised learning (SSL). SSL models are firstly pretrained on a large amount of unlabeled examples and then finetuned on task-specific data. This paradigm has been particularly successful for natural language processing (Peters et al., 2018; Devlin

[1] Audio samples are available at: `https://neurtts.github.io/prosodybert_demo/`.

et al., 2019). Recent speech SSL models like wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and WavLM (Chen et al., 2022) achieve good performance on speech recognition and understanding tasks, especially when only small-amount of task data is available. However, these methods focus on phone-level phonetic information and filter out long-range prosodic information. (Polyak et al., 2021; Weston et al., 2021). Self-supervised methods has been explored for prosody learning. However, these approaches still rely on resources other than raw audio. For example, Polyak et al. (2021) requires a pretrained speech-to-unit model (e.g. HuBERT (Hsu et al., 2021)) to extract content information; Weston et al. (2021) needs the word boundary information from transcripts to learn word-level prosodic information.

To address the above challenges, we propose ProsodyBERT, a self-supervised learning method that disentangles prosody features from speech content and speaker information. Similar to HuBERT Hsu et al. (2021), we pretrain an SSL model by masked unit prediction. The pseudo labels are given by K-means clustering on speaker-normalized acoustic prosody attributes (pitch, energy, their dynamics, etc.), which encourages the model to focus on prosody learning. High-frequency information is removed in speech input to disentangle our prosody feature with lexical content. Finally, inspired by SpanBERT (Joshi et al., 2020), we propose a span boundary loss to encourage the model to better represent long-range prosody information. We also radically compress the model size and reduce the feature dimensions to make our model easier to use. Similar to prior SSL models, ProsodyBERT is firstly trained on a large amount of raw speech audio and then adapted to target speakers. Such a design enables ProsodyBERT to learn the full distribution of prosody on massive amounts of low-quality data without relying on any transcripts or external models. It can be used as a general prosody feature extractor, replacing prior features like energy and pitch.

We demonstrate the effectiveness of pretrained ProsodyBERT features on text-to-speech (TTS) and emotion recognition (ER). During TTS training, we extract ProsodyBERT features from speech and use them as conditional inputs for the TTS decoder. A separate prosody predictor is trained such that it takes text and style as inputs and generates prosody features. During TTS inference, the TTS decoder takes the predicted prosody features as input. Experiments show that the TTS system trained with ProsodyBERT features can generate natural and expressive speech, surpassing FastSpeech 2 (Ren et al., 2020) (trained with energy and pitch) by a large margin on subjective human evaluation. Furthermore, the style and expressiveness can be controlled by prosody features. For example, given the prosody features of reading speech style and spontaneous speech style, by summing these features with different weights, and using them as the input for the TTS decoder, we can synthesize speech with different level of expressiveness. We also conduct analysis on the de-identifiablility of ProsodyBERT features. For ER task, we simply concatenate ProsodyBERT features to HuBERT features and use them as the input for downstream models. We achieve new state-of-the-art on IEMOCAP emotion recognition task, showing that ProsodyBERT features are complementary to HuBERT features.

2 Related Work

Modeling Prosody in TTS and Voice Conversion Most prior works on prosody treat prosody feature learning as an auxiliary module for downstream generation tasks. Recent approach includes directly using signal prosody (F_0, energy, etc.) (Valle et al., 2020; Ren et al., 2020; Kenter et al., 2020; Liu et al., 2021; Kharitonov et al., 2022), learning a latent style embedding (Wang et al., 2018; Zhang et al., 2019; Hsu et al., 2018; Sun et al., 2020), learning frame-level or phone-level representation (Du & Yu, 2021; Kreuk et al., 2021), and utilize reference audios for style Choi et al. (2020); Yi et al. (2022). Most of these prosody representations rely on task-specific models.

Learning Prosody Representation Prior works on prosody representation learning are mainly based on information bottleneck and self-supervised learning. Information bottleneck approaches rely on carefully designed bottlenecks that condition on lexical and timbre information to capture the residual prosody information. (Qian et al., 2020; Ren et al., 2022). Polyak et al. (2021) uses pretrained speech-to-unit models (e.g. CPC van den Oord et al. (2018), HuBERT Hsu et al. (2021)) to extract lexical information in a self-supervised way and learn disentangled prosody representation. Weston et al. (2021) learns self-supervised prosody features via a contrastive task similar to Baevski et al. (2020). However, it relies on word boundaries provided by transcripts to learn word-level prosody representation.

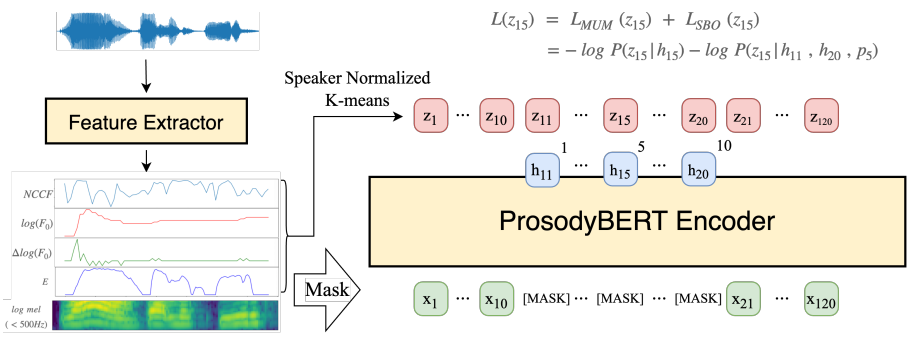


Figure 1: Overview of ProsodyBERT pretraining. Given a raw audio piece, signal processing tools extract the acoustic-prosodic features (NCCF, F_0, energy, deltas) and the log Mel spectrum (with bins $< 500Hz$). Offline K-means clustering is done on the speaker-normalized acoustic-prosodic features of all the frames in the pretraining corpus to generate the hidden cluster assignments ($z_{11}, z_{12}, ..., z_{20}$) as the prediction targets. The equation shows the loss terms on the masked frame x_{15}. The masked unit modeling (MUM) objective uses the corresponding model outputs h_{15} to predict z_{15}. The span boundary objective (SBO) uses the model outputs on the boundary to predict the cluster assignments of every frame in the masked span. Here z_{15} is predicted given h_{11}, h_{20}, and the relative position embedding p_5.

3 PROSODYBERT

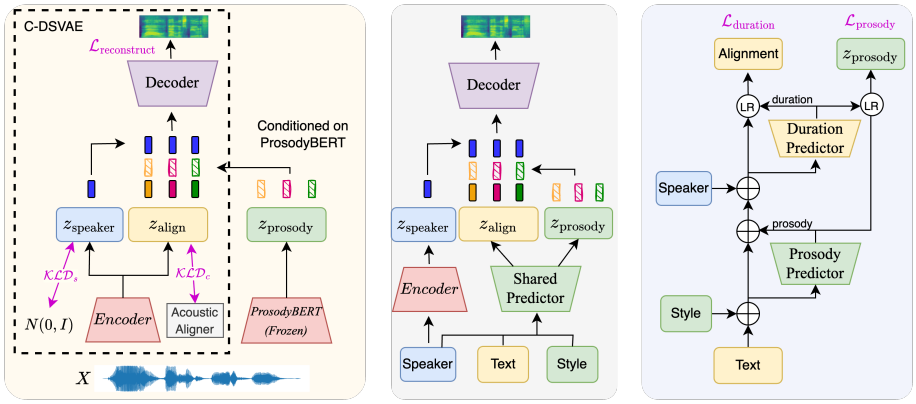
An overview of ProsodyBERT model is given in Figure 1. Our model is based on HuBERT Hsu et al. (2021) and SpanBERT Joshi et al. (2020), but modified to focus on prosody representation learning and disentangle prosody from lexical content and speaker information.

Notation Given an audio segment A, the feature extractor outputs a sequence of acoustic features $X = (x_1, x_2, \cdots, x_n)$, each corresponding to a fixed-length frame. x_i are log Mel spectral features concatenated with acoustic-prosodic features. n corresponds to the number of frames. ProsodyBERT takes X as input and produces a contextualized vector representation for each frame $(h_1, h_2, \cdots, h_n)$.

Feature Extractor Given the raw audio segment A, ProsodyBERT feature extractor first conducts loudness normalization on A to 0dB, and then computes prosodic and spectral features. All features are frame-level with a fixed frame length. For prosodic features, we include log fundamental frequency ($\log(F_0)$), Normalized Cross Correlation Function (NCCF), energy, and $\Delta\log(F_0)$. F_0 and NCCF are extracted via the Kaldi pitch tracker (Povey et al., 2011). Pitch ($\log(F_0)$) and energy (E) are globally normalized on corpus-level by z-score, and the deltas are computed on the normalized values. No normalization is done on NCCF features. For spectral features, we take the low-frequency bands (first 20 bins, $< 500Hz$) of log Mel spectrum as additional input for ProsodyBERT. The feature extractor filters out all the high-frequency information in the audio input, discarding the formants that mainly contain lexical content(Ren et al., 2022; Weston et al., 2021). Mean and variance normalization are performed on these low-frequency bins on utterance level. $\log(F_0)$, E , NCCF, $\Delta\log(F_0)$, and log Mel features are concatenated at each frame as the output X.

Hidden Units for ProsodyBERT Inspired by HuBERT Hsu et al. (2021), we get the frame-level target labels by acoustic unit discovery. We first train a K-means clustering model on signal prosody extracted by our feature extractor, i.e. $(\text{NCCF}, \log(F_0), \Delta\log(F_0), E)$ on all the frames in the pretraining corpus. Z-score speaker normalization is done on E and $\log(F_0)$ before clustering to disentangle speaker information. For a speech utterance with acoustic features $X = (x_1, x_2, \cdots, x_n)$, the discovered acoustic units are $Z = (z_1, z_2, \cdots, z_n)$, where $z_t \in [C]$ is a C-class categorical variable. C is the number of clusters of the pre-trained K-means model.

Training objectives We adopt the same masking mechanism as SpanBERT (Joshi et al., 2020), wav2vec 2.0 (Baevski et al., 2020), and HuBERT (Hsu et al., 2021). Let the input utterance be $X = (x_1, x_2, \cdots, x_n)$. Denote the set of all masked frames as $M \subset X$. Define $\tilde{X}$ as the corrupted input sequence, in which the frames in M are replaced with a special mask embedding. ProsodyBERT



(a) UTTS training (b) UTTS inference (c) Prosody and duration predictor training

Figure 2: An overview of UTTS conditioned on ProsodyBERT features. The loss terms during training are marked purple.

has two training objectives. The first is the masked unit prediction objective. Let the output hidden states at the tth frame be h_t. We compute the masked unit modeling loss of frame t as:

$$L_{\text{MUM}(z_t)} = -\log\ p_m(z_t|h_t) \tag{1}$$

The masked prediction model p_m takes h_t as input and predicts a distribution over target label z_t. It is implemented as a single-layer perceptron followed by softmax.

Previous work mostly relies on alignments to learn the span-level representation of speach (Weston et al., 2021; Hu et al., 2021). In contrast, our self-supervised method only relies on raw audio. Inspired by SpanBERT (Joshi et al., 2020), we add a span boundary loss to encourage ProsodyBERT to learn long-range prosodic representation. Given a masked span $(x_s, \cdots, x_e)$, in which s and e are the start and ending frames of the span, and let x_t be a frame in this span. The span boundary loss on x_t is computed as:

$$L_{\text{SBO}}(z_t) = -\log\ p_{\text{span}}(z_t|h_s, h_e, p_{t-s}, q_{e-t+1}) \tag{2}$$

in which h_s, h_e are the output hidden states of the span boundaries x_s, x_e, p, q are the relative positional embedding with respect to the left boundary x_s and the right boundary x_e, respectively. The span prediction model p_{span} is implemented as a 2-layer feedforward network followed by softmax. The span boundary loss forces the model to predict the entire masked span without relying on individual tokens within it. The total loss is computed over all the masked frames:

$$L(\tilde{X}, M, Z) = \sum_{t \in M} (L_{\text{MUM}}(z_t) + L_{\text{SBO}}(z_t)) \tag{3}$$

4 ProsodyBERT for Style-Controllable TTS

4.1 Background: UTTS

We demonstrate the effectiveness of ProsodyBERT on UTTS Lian et al. (2022b). An overview of UTTS training and inference is shown in Figure 2. UTTS is an unsupervised multi-speaker TTS framework that does not require text-audio pairs for the TTS acoustic modeling. It is developed from the perspective of disentangling speech representation learning. We choose UTTS because it is an unsupervised TTS framework that could benefits from pretraining on large amount of data, which aligns with ProsodyBERT's self-supervised design. Also, our initial experiments indicate that UTTS synthesizes better quality speech than the other trained models provided by Lee et al. (2022).

Baseline UTTS Training Speaker and alignment representation is disentangled via a conditional disentangled sequential variational autoencoder (C-DSVAE) Lian et al. (2022a) during self-supervised training. Let a training speech instance be X. Denote the model parameters as θ. Let p_θ be the prior model, and q_θ be the posterior model. The training objectives of C-DSVAE are as follows:

$$\mathcal{L}_{\text{KLD}_s} = \mathbb{E}_{p(X)}[\text{KLD}(q_\theta(z_{\text{speaker}}|X)||N(0,I))] \tag{4}$$

$$\mathcal{L}_{\text{KLD}_c} = \mathbb{E}_{p(X)}[\text{KLD}(q_\theta(z_{\text{align}}|X)||p_\theta(z_{\text{align}})] \tag{5}$$

$$\mathcal{L}_{\text{reconstruct}} = \mathbb{E}_{p(X)}\mathbb{E}_{q_\theta(z_{\text{speaker}},z_{\text{align}}|X)}[-\log(p_\theta(X|z_{\text{speaker}}, z_{\text{align}}))] \tag{6}$$

Here KLD is the KL-divergence. $z_{\text{speaker}}, z_{\text{align}}$ corresponds to the hidden states of the speaker and the phone alignment. $\mathcal{L}_{\text{reconstruct}}$ is the reconstruction loss. $\mathcal{L}_{\text{KLD}_s}$ and $\mathcal{L}_{\text{KLD}_c}$ forces disentanglement between z_{speaker} and z_{align}. Here z_{speaker} is utterance-level and z_{align} is frame-level.

4.2 Condition UTTS on ProsodyBERT features

However, an issue of the UTTS framework is that only speaker and textual content (phone alignment) information is kept during training. The prosody variations are lost, making the synthesized demos less expressive and unnatural. ProsodyBERT provides a solution to this issue. As shown in Figure 2 (a), during training, we condition the TTS decoder with the prosody features extracted from the speech segment X. Formally, we replace Equation 6 into:

$$z_{\text{prosody}} = \text{ProsodyBERT}(X)^2 \tag{7}$$

$$\mathcal{L}_{\text{reconstruct}} = \mathbb{E}_{p(X)}\mathbb{E}_{q_\theta(z_{\text{speaker}},z_{\text{align}}|X)}[-\log(p_\theta(X|z_{\text{speaker}}, z_{\text{align}}, z_{\text{prosody}}))] \tag{8}$$

ProsodyBERT is designed to focus on prosody and disentangle textural content and speaker identity. Thus, by adding prosody features as the third source of reconstruction, the prosodic variations are accounted by these prosody features (z_{prosody}), and the C-DSVAE can focus on learning disentangled representation of the speaker (z_{speaker}) and content (z_{align}). Notice that the pretrained ProsodyBERT is frozen during TTS training.

Phone-level and word-level prosody features Due to the high variance of prosody, it is hard to predict the frame-level prosody features directly. (Ren et al., 2020) Instead, given the forced alignments of the training text, we conduct a mean-pooling of the frame-level prosody features according to the phone/word boundaries and replace the frame-level features with these mean-pooling values at each frame. We denote these features as the phone-level/word-level prosody features and use them in both TTS training and inference. Comparisons between phone-level and word-level features are given in Section 6.1.1.

4.3 UTTS Inference

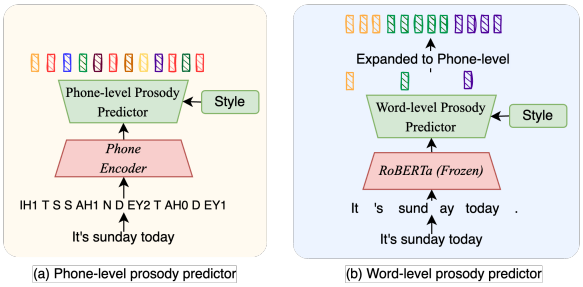


Figure 3: Diagrams of phone-level prosody predictor and word-level prosody predictor.

An overview of the UTTS inference pipeline is in Figure 2 (b). Target speaker, text, and style ID are mapped into the learned representations of C-DSVAE to utilize the trained decoder for TTS. Lian et al. (2022b) has explained the process of mapping a target speaker to z_{speaker}, and a phone alignment to unsupervised alignment z_{align}. Here we focus on predicting the prosody features and phone alignment, which is illustrated in Figure 2 (c). The predictor first predicts the prosody features

[2]Notice that here we use a slightly different version of frame-level ProsodyBERT features, as discussed in the paragraph "phone-level and word-level prosody features".

given the text and the style and then predict the duration of each phone given the style, the prosody features, and the speaker. Each phone is expanded according to the predicted duration to form the phone alignment.

Prosody Predictor Training Details of the prosody predictor are shown in Figure 3. We have two configurations, i.e., predict prosody features at phone-level and word-level. The phone-level predictor takes the phone sequence and a style embedding as input, and predicts the phone-level prosody features for each phone. The phone encoder, the style embedding, and the prosody predictor are trained. The word-level predictor follows the design of Kenter et al. (2020). The input text is tokenized into WordPieces and passes through a frozen RoBERTa encoder. Since the TTS system has no notion of WordPiece, the embedding corresponding to the first WordPiece of each word is used as the word representation. A prosody predictor takes the RoBERTa embeddings and the style embedding as input, and predicts the word-level prosody features for each word.

During training, for both kinds of predictor, suppose given an input text, the model outputs are $O = (o_1, o_2, \cdots, o_k)$, the and the target ProsodyBERT features as $H = (h_1, h_2, \cdots, h_k)$. Here k refers to the number of phones for phone-level predictor, and number of words for word-level predictor. Our training loss is:

$$\mathcal{L}_{\text{prosody}} = \sum_{t=1}^{k} [\frac{1}{D} ||o_t - h_t||_1 - \log \sigma(\cos(o_t, h_t))] \tag{9}$$

Here D is the dimension of hidden states, σ is the sigmoid function, and $\cos$ is the cosine similarity. This loss is proposed in (Chang et al., 2022), and we empirically find it yields good performance.

Duration Predictor The duration predictor takes the phoneme sequence, the predicted prosody features (word-level features are expanded to phone-level via a lexicon), the style embedding, and the speaker embedding as inputs, and predicts the duration (number of frames) of each phoneme. To support the zero-shot scenarios for speakers, we use the speaker embedding from pretrained ECAPA-TDNN (Desplanques et al., 2020; Zhang & Yu, 2022). Montreal Forced Alignment (MFA) (McAuliffe et al., 2017) is used to extract the target phoneme duration, and the training loss is the mean square error (MSE) between predicted duration and target duration. All durations are transformed to the logarithmic domain for ease of training. During inference, given the predicted duration, the input phoneme sequence is broadcasted to frame-level and become the predicted forced alignment. The predicted prosody features are broadcasted in the same way to the frame-level features z_{prosody}.

5 Experimental Setup

5.1 ProsodyBERT Pretraining

We use the full LibriTTS (Zen et al., 2019) audio for ProsodyBERT pretraining. LibriTTS contains 586 hours of audiobook data from 2,456 speakers. To generate the target labels, we run K-means clustering with 100 clusters using the `MiniBatchKMeans` algorithm in `scikit-learn` (Pedregosa et al., 2011). The mask span length is set to $l = 10$ for $20ms$ frames, and $l = 20$ for $11ms$ frames. Notice that the masks can overlap, so the real span length is variable. We choose this span length because the average word duration in human speech is about 200-300ms. For the sampling of mask starting point, we set the probability to be 65%, resulting in about 50% of frames being masked. We radically reduce the model size and feature dimensions to make ProsodyBERT easier to use. ProsodyBERT has two configurations. For ER, following (Hsu et al., 2021), the input audio sampling rate is 16kHz, and the frameshift is 20ms. For TTS, following (Ren et al., 2020), the input sampling rate is 22.05kHz, and the hop length is 256 (frame shift $\sim$ 11ms). ProsodyBERT architecture follows the distilBERT architecture (Sanh et al., 2019), while the positional embeddings are replaced with the convolutional positional embedding as in (Baevski et al., 2020) to support long inputs. The encoder has 6 transformer layers, with hidden dimension 512, 8 heads, feedforward dimension 2048. The final projection layer has dimension 32. The model has 21M parameters. We use Adam (Kingma & Ba, 2014) optimizer and linear learning rate schedule, with peak learning rate $1e-4$. For both SLU and TTS version, we train the model for 10 iterations on 8 GPUs, with batch size 8. Training takes about 6 hours for SLU version and 10 hours for TTS version (due to shorter frame shift).

5.2 EXPERIMENTS ON TEXT-TO-SPEECH (TTS)

For text-to-speech experiments, two datasets are used: VCTK (Veaux et al., 2017) and DailyTalk (Lee et al., 2022). VCTK contains 44 hours of read speech from 109 speakers and does not contain much prosody variation. DailyTalk contains 21.6 hours of spontaneous dialogue from 2 speakers and has rich prosody. We pool these two datasets to adapt our TTS system to support multi-speakers (even zero-shot speaker) and two styles. VCTK is treated as reading speech style, and DailyTalk is treated as spontaneous speech style. The style embeddings are learned during training. The architecture of UTTS follows Lian et al. (2022b). The architecture of prosody and duration predictor follows the variance adaptor of Ren et al. (2020).

5.3 EXPERIMENTS ON EMOTION RECOGNITION (ER)

We conduct the emotion recognition experiments on IEMOCAP Busso et al. (2008) dataset. The dataset contains 5 sessions. We follow the "leave-one-session-out" setting. In each round, one session is used for test and the other sessions are used as train and validation sets. The evaluation metric is weighted accuracy, which is the accuracy of all utterances in the test session. There are two experiment settings. The first is SUPERB (Yang et al., 2021) probing setting, in which the pretrained speech models are freezed. The downstream model takes the weighted average of each model layer as the input. For these experiment we concatenate our 32-dim ProsodyBERT features to the input. All experiments in SUPERB setting are conducted with S3PRL toolkit provided with SUPERB. We also explore the best-possible results of our proposed prosody features with ESPnet (Watanabe et al., 2018; Arora et al., 2021). The model utilizes the pre-trained model HuBERT and our ProsodyBERT as a feature extractor, following with a conformer-based encoder-decoder module. The model is jointly trained with an auto-regressive setting, which predicts the emotion label and the corresponding transcription from the given utterance. To maximize the model performance, we optimize the whole model, including the pre-trained model and our ProsodyBERT.[3]

6 RESULTS AND ANALYSIS

6.1 CONTROLLABLE-TTS

6.1.1 MEAN OPINION SCORE (MOS) TESTS

	MOS(↑)	WER (↓)
Ground truth	4.53	5.6%
FastSpeech 2	3.37	25.3%
Baseline UTTS	3.50	23.2%
Phone prosody	3.63	21.1%
Word prosody	3.81	18.9%

Table 1: MOS scores and ASR word error rate (WER) of different TTS settings on DailyTalk.

Representation	EER
Energy + Pitch + low freq mel	8.2%
ProsodyBERT	35.3%

Table 2: Equal Error Rate (EER) of speaker verification on VCTK test set, using different prosody representations. Higher EER implies higher level of de-identification.

The results on Mean Opinion Score (MOS) tests on DailyTalk are shown in Table 1. 20 subjects are asked to evaluate the naturalness of synthesized speech. We compare our systems with the ground truth and the official FastSpeech 2 checkpoint. Lee et al. (2022); Ren et al. (2020). The FastSpeech2 model uses energy and pitch as the supervision signal to improve the prosody of synthesized audio. The UTTS systems follows the experiment settings we discused in Section 5. MOS scores are shown in Table 1. We also conduct intelligibility evaluation using a LibriSpeech ASR model[4] on DailyTalk validation set. The trend of WER is consistent with the MOS scores. Overall, UTTS systems that are trained with ProsodyBERT features are viewed much better compared with the baseline UTTS

[3]The hyper-parameters for training and decoding are the same as the default ESPnet recipe at `https://github.com/espnet/espnet/tree/master/egs2/iemocap/asr1`

[4]`https://huggingface.co/espnet/simpleoier_librispeech_asr_train_asr_conformer7_hubert_ll60k_large_raw_en_bpe5000_sp`

and the FastSpeech 2. Also, the word-level prosody predictor is considered better than phone-level prosody predictor.

Detailed comparison of synthesized speech We compare the synthesized audios and the ground truth in detail on Figure 4. The baseline UTTS system fails to predict the prosody contour of the ground truth speech. FastSpeech 2, which is conditioned on energy and F_0 during training, also fails. Meanwhile, the speech synthesized by UTTS systems conditioned on phone-level and word-level ProsodyBERT features has highly similar prosody contours compared with the ground truth audio. These results demonstrate that self-supervised ProsodyBERT features are much easier for prosody modeling than signal prosody features like energy and F_0 , especially when only small amount of high-quality TTS data is available.

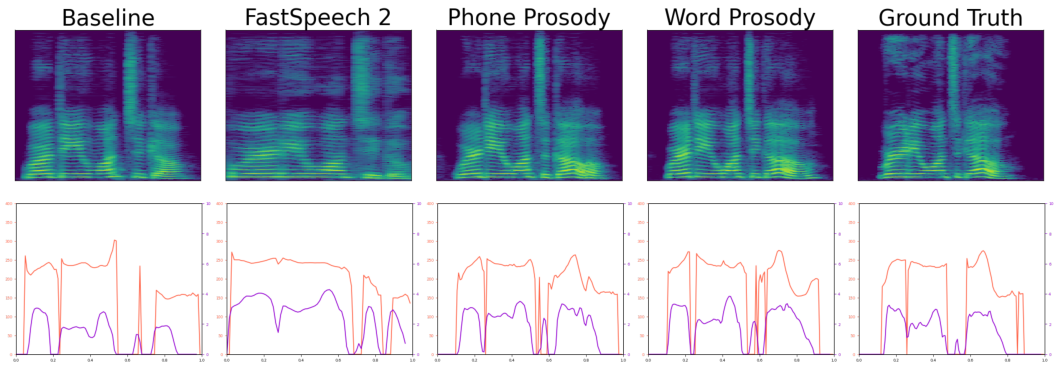


Figure 4: Comparison of the log Mel spectrograms, pitch, and energy contours of synthesized speech "where do you want to go?" and the ground truth in DailyTalk (5_1_d1127). The orange line is F_0 and the puple line is E.

6.1.2 Expressiveness Control

As discussed in Section 5, our system has two styles: VCTK style (read speech) and DailyTalk style (spontaneous speech). However, in practice, people might consider VCTK style too flat, and DailyTalk style too dramatic. We control the level of expressiveness by weighted sum of the word-level prosody features with different weights. The number in the figure titles are the weight of VCTK style prosody features. As the weight become bigger, the synthesized speech become more flat. Also, human evaluation suggests that the naturalness of synthesized speech is kept for all the weights.

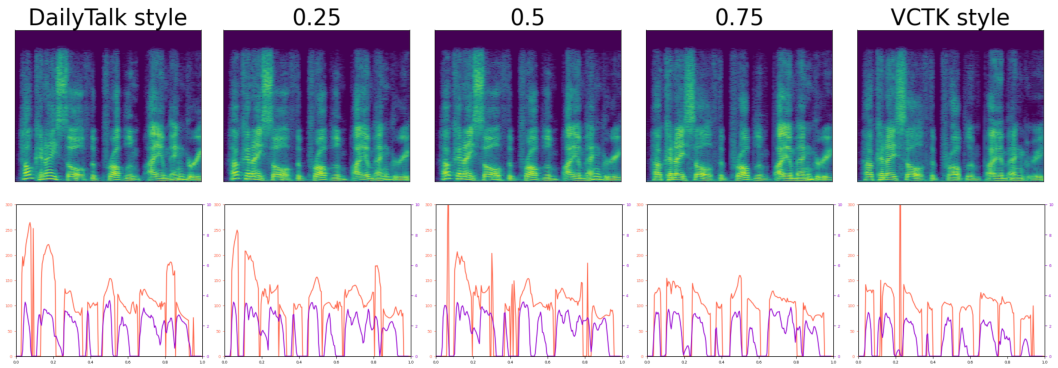


Figure 5: The log Mel spectrograms, pitch, and energy contours conditioned on different level of expressiveness, controlled by weights of ProsodyBERT features of "VCTK style" and "DailyTalk style". The text is "how can I solve the problem? I am angry."

6.2 DE-IDENTIFICATION

We compare the de-identification of ProsodyBERT with other prosody representations via a speaker verification task. The task is performed on 36900 randomly generated trails from VCTK test set with the same setting as Lian et al. (2022c;a;b). The baseline is energy + pitch features + low-frequency bins of log Mel spectrogram. We perform mean pooling of the frame-level prosody representations as the vector representation of each utterance. Higher Equal Error Rate (EER) implies less speaker related information in the features. Table 2 shows that ProsodyBERT features has higher level of de-identification than acoustic-prosodic features (energy, pitch, and low-freq mel).

6.3 EMOTION RECOGNITION ON IEMOCAP

Model	Acc.	
	Original	+ProsodyBERT
Wav2vec2-base	63.4	64.2
Wav2vec2-large	65.6	68.9
Data2vec-base	66.3	70.4
Data2vec-large	66.3	70.3
HuBERT-base	64.9	66.1
HuBERT-large	67.6	69.1

Table 3: Results on IEMOCAP dataset in SUPERB (Yang et al., 2021) probing setting.

Method	Acc.
WISE (Shen et al., 2020)	66.5
wave2vec2-PT (Pepino et al., 2021)	67.2
HuBERT-large (Gat et al., 2022)	71.9
HuBERT-large + TAP (Gat et al., 2022)	74.2
Ours	
HuBERT-large + Signal Prosody	74.3
HuBERT-large + ProsodyBERT	**75.8**

Table 4: Results on IEMOCAP dataset in supervised setting. The “signal prosody" refers to ProsodyBERT inputs, i.e. pitch, energy, NCCF, deltas, and log mel.

Table 3 shows our results on IEMOCAP in SUPERB probing setting, in which the pretrained models are all frozen during training. ProsodyBERT features are concatenated to the existing self-supervised models as the inputs for downstream models. We experiment with wav2vec 2.0 (Baevski et al., 2020), Data2vec (Baevski et al., 2022), and HuBERT (Hsu et al., 2021), and find consistent improvement after adding prosody features. These results demonstrate that ProsodyBERT captures additional prosodic information that are complementary to these pretrained models.

Table 4 compares our results with the current state-of-the-arts on IEMOCAP in fully supervised setting, in which all parameters are finetuned. Notice that we also perform domain-adaptive pretraining on the training sessions of IEMOCAP. Our method outperforms the current state-of-the-art (Gat et al., 2022) on this task. For analysis, we compare the ER accuracy of "HuBERT + ProsodyBERT" with "HuBERT + Signal Prosody" features, which includes pitch, energy, NCCF, deltas, and log mel. The relative performance gap shows the benefit of self-supervised learning.

7 CONCLUSION

We propose ProsodyBERT, a self-supervised method to learning prosody representations apart from speech content and speaker information. Our method does not rely on any transcripts or external models. This self-supervised framework allows ProsodyBERT to learn the full distribution of prosody variations via pretraining on large amount of data. We demonstrate the effectiveness of ProsodyBERT on a style-controllable TTS system, showing that the TTS model trained with ProsodyBERT features can generate natural and expressive speech, outperforming the model trained with energy and pitch. We also achieve new state-of-the-art on IEMOCAP emotion recognition, showing that ProsodyBERT features are complementary to pretrained models like HuBERT, wav2vec2, and data2vec. Future work may explore broader usage of ProsodyBERT. For example, ProsodyBERT may be combined with large language model on spoken language understanding tasks by providing prosody information that is not covered by text.

8 REPRODUCIBILITY STATEMENT

The model details are given in Section 5. The source code for the model architecture, together with all the hyperparameters, are provided in the supplementary material. The trained model checkpoints will be released after the anonymity period, and they can be directly used with the provided source code. For the training procedure, we have accounted for the hardest issues in Appendix. The TTS demos uses in the MOS test are also provided in the supplementary material for reference.

REFERENCES

Siddhant Arora, Siddharth Dalmia, Pavel Denisov, Xuankai Chang, Yushi Ueda, Yifan Peng, Yuekai Zhang, Sujay Kumar, Karthik Ganesan, Brian Yan, et al. Espnet-slu: Advancing spoken language understanding through espnet. *arXiv preprint arXiv:2111.14706*, 2021.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language, 2022.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

Heng-Jui Chang, Shu-wen Yang, and Hung-yi Lee. Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022. doi: 10.1109/icassp43922.2022.9747490. URL http://dx.doi.org/10.1109/icassp43922.2022.9747490.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–14, 2022. ISSN 1941-0484. doi: 10.1109/jstsp.2022.3188113. URL http://dx.doi.org/10.1109/jstsp.2022.3188113.

Jenny Yeonjin Cho, Sara Ng, Trang Tran, and Mari Ostendorf. Leveraging prosody for punctuation prediction of spontaneous speech. *Interspeech 2022*, 2022.

Seungwoo Choi, Seungju Han, Dongyoung Kim, and Sungjoo Ha. Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding. *Interspeech 2020*, Oct 2020. doi: 10.21437/interspeech.2020-2096. URL http://dx.doi.org/10.21437/interspeech.2020-2096.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pp. 3830–3834, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Chenpeng Du and Kai Yu. Rich prosody diversity modelling with phone-level mixture density network. *Interspeech 2021*, Aug 2021. doi: 10.21437/interspeech.2021-802. URL http://dx.doi.org/10.21437/interspeech.2021-802.

Itai Gat, Hagai Aronowitz, Weizhong Zhu, Edmilson Morais, and Ron Hoory. Speaker normalization for self-supervised speech emotion recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7342–7346, 2022. doi: 10.1109/ICASSP43922.2022.9747460.

Wei-Ning Hsu, Yu Zhang, Ron J. Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang. Hierarchical generative modeling for controllable speech synthesis, 2018.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. ISSN 2329-9304. doi: 10.1109/taslp.2021.3122291. URL http://dx.doi.org/10.1109/taslp.2021.3122291.

Yushi Hu, Shane Settle, and Karen Livescu. Acoustic span embeddings for multilingual query-by-example search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 935–942, 2021. doi: 10.1109/SLT48900.2021.9383545.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, Dec 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00300. URL http://dx.doi.org/10.1162/tacl_a_00300.

Tom Kenter, Manish Kumar Sharma, and Rob Clark. Improving prosody of rnn-based english text-to-speech synthesis by incorporating a bert model. In *INTERSPEECH 2020*, 2020.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Morgane Riviere, Abdelrahman Mohamed, Emmanuel Dupoux, and Wei-Ning Hsu. Text-free prosody-aware generative spoken language modeling. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022. doi: 10.18653/v1/2022.acl-long.593. URL http://dx.doi.org/10.18653/v1/2022.acl-long.593.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Felix Kreuk, Adam Polyak, Jade Copet, Eugene Kharitonov, Tu-Anh Nguyen, Morgane Rivière, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, and Yossi Adi. Textless speech emotion conversion using discrete and decomposed representations, 2021.

Keon Lee, Kyumin Park, and Daeyoung Kim. Dailytalk: Spoken dialogue dataset for conversational text-to-speech, 2022.

Jiachen Lian, Chunlei Zhang, Gopala Krishna Anumanchipalli, and Dong Yu. Towards improved zero-shot voice conversion with conditional dsvae. In *Interspeech*, 2022a.

Jiachen Lian, Chunlei Zhang, Gopala Krishna Anumanchipalli, and Dong Yu. Utts: Unsupervised tts with conditional disentangled sequential variational auto-encoder, 2022b.

Jiachen Lian, Chunlei Zhang, and Dong Yu. Robust disentangled variational speech representation learning for zero-shot voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6572–6576. IEEE, 2022c.

Rui Liu, Berrak Sisman, Guanglai Gao, and Haizhou Li. Expressive tts training with frame and style reconstruction loss. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 1806–1818, 2021. ISSN 2329-9304. doi: 10.1109/taslp.2021.3076369. URL http://dx.doi.org/10.1109/TASLP.2021.3076369.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *INTERSPEECH*, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Leonardo Pepino, Pablo Riera, and Luciana Ferrer. Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv preprint arXiv:2104.03502*, 2021.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. *Interspeech 2021*, Aug 2021. doi: 10.21437/interspeech.2021-475. URL http://dx.doi.org/10.21437/interspeech.2021-475.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson. Unsupervised speech decomposition via triple information bottleneck, 2020.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

Yi Ren, Ming Lei, Zhiying Huang, Shiliang Zhang, Qian Chen, Zhijie Yan, and Zhou Zhao. Prosospeech: Enhancing prosody with quantized vector pre-training in text-to-speech. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022. doi: 10.1109/icassp43922.2022.9746883. URL http://dx.doi.org/10.1109/icassp43922.2022.9746883.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019.

Guang Shen, Riwei Lai, Rui Chen, Yu Zhang, Kejia Zhang, Qilong Han, and Hongtao Song. Wise: Word-level interaction-based multimodal fusion for speech emotion recognition. In *Interspeech*, pp. 369–373, 2020.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pp. 4693–4702. PMLR, 2018.

Guangzhi Sun, Yu Zhang, Ron J. Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu. Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9053520. URL http://dx.doi.org/10.1109/ICASSP40776.2020.9053520.

Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf. Parsing speech: A neural approach to integrating lexical and acoustic-prosodic information, 2017.

Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9054556. URL http://dx.doi.org/10.1109/ICASSP40776.2020.9054556.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2017. URL https://doi.org/10.7488/ds/2645.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis, 2018.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pp. 2207–2211, 2018. doi: 10.21437/Interspeech.2018-1456. URL http://dx.doi.org/10.21437/Interspeech.2018-1456.

Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. Learning de-identified representations of prosody from raw audio, 2021.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. Superb: Speech processing universal performance benchmark. *Interspeech 2021*, Aug 2021. doi: 10.21437/interspeech.2021-1775. URL http://dx.doi.org/10.21437/interspeech.2021-1775.

Yuanhao Yi, Lei He, Shifeng Pan, Xi Wang, and Yujia Xiao. Prosodyspeech: Towards advanced prosody model for neural text-to-speech. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7582–7586, 2022. doi: 10.1109/ICASSP43922.2022.9746744.

Vicky Zayats, Trang Tran, Richard Wright, Courtney Mansfield, and Mari Ostendorf. Disfluencies and human speech transcription errors. *Interspeech 2019*, Sep 2019. doi: 10.21437/interspeech.2019-3134. URL http://dx.doi.org/10.21437/interspeech.2019-3134.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *Interspeech 2019*, Sep 2019. doi: 10.21437/interspeech.2019-2441. URL http://dx.doi.org/10.21437/interspeech.2019-2441.

Chunlei Zhang and Dong Yu. C3-dino: Joint contrastive and non-contrastive self-supervised learning for speaker verification. *IEEE Journal of Selected Topics in Signal Processing*, 2022.

Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. Learning latent representations for style control and transfer in end-to-end speech synthesis. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019. doi: 10.1109/icassp.2019.8683623. URL http://dx.doi.org/10.1109/ICASSP.2019.8683623.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Emotional voice conversion: Theory, databases and esd. *Speech Communication*, 137:1–18, Feb 2022. ISSN 0167-6393. doi: 10.1016/j.specom.2021.11.006. URL http://dx.doi.org/10.1016/j.specom.2021.11.006.

Appendices

A Overview of data used

Dataset	Domain	Hours	Total speakers	Usage
LibriTTS (Zen et al., 2019)	Audiobook	586	2,456	ProsodyBERT pretraining
VCTK (Veaux et al., 2017)	Reading	44	109	Text-to-speech
DailyTalk (Lee et al., 2022)	Conversation	21.6	2	Text-to-speech
IEMOCAP (Busso et al., 2008)	Conversation	12.5	10	Emotion Recognition

Table 5: Summary of the datasets we used.

An overview of the data we use are shown on Table 5.

B More on De-Identification

B.1 Motivation

Our framework disentangles ProsodyBERT features from speaker information. The TTS model does not takes speaker information as input when predicting prosody. The main motivation of de-identification is for the robustness and generalizability of the controllable TTS system, especially in zero-shot speaker scenario. Suppose the prosody features are not de-identified. Then the TTS model needs to take speaker information as input to predict the prosody feature. However, for zero-shot speaker, due to the lack of training, the model might generate the prosody of a random speaker, which may cause mismatch when combining with the zero-shot speaker's timbre information. Our experiments and demos show that it is possible to predict universal prosody features for all the speakers, taking the text and style as input. These speaker-independent prosody features is more suitable for controllable TTS systems and allows better generalization for zero-shot speakers.

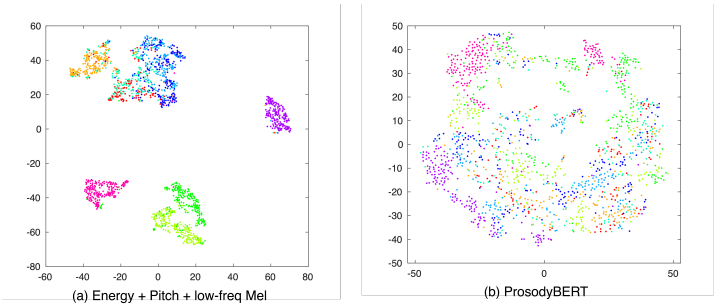


Figure 6: Visualization of mean-pooled prosody representations on the utterances in VCTK test set. Each color represents a speaker.

B.2 Visualization of prosody features

Figure 6 shows the t-SNE visualization of the utterance prosody representations in VCTK test set, computed as described in 6.2. The baseline is the signal prosody features (energy + pitch features + low-frequency bins of log Mel spectrogram). Each color represents a speaker. The plot shows that the baseline signal has strong speaker information, and the utterances of each speaker is clustered. Meanwhile, the ProsodyBERT features are spreaded in the hidden space, showing a higher-level of de-identification.

A IMPLEMENTATION DETAILS

Span Masking We adopt the same masking mechanism as SpanBERT (Joshi et al., 2020), wav2vec 2.0 (Baevski et al., 2020), and HuBERT (Hsu et al., 2021). Let the input utterance be $X = (x_1, x_2, \cdots, x_n)$, we first randomly select a subset of $Y \subseteq X$ such that $|Y| = m|X|$. m is a given hyperparameter, and it is set to 65% in our experiments. The frames in Y are the starting point of the masked spans, and spans of l frames are masked. Notice that the spans can overlap, so the length of the spans is not fixed.

Alignments of WordPiece, Phonemes, and Words There is no direct mapping between WordPieces and Phones. However, the RoBERTa encoder in word-level prosody predictor only takes WordPiece as inputs, and the UTTS system takes frame-level inputs. To deal with this mapping issue, we first map words to WordPieces (represent each word by its first WordPiece, and ignore the other WordPieces), and then broadcast this word-level feature to its corresponding phones via a lexicon.