ExpressTTS: Augmentation for Speech Recognition with Expressive Speech Synthesis

Lindsay Kempen[1]

[1]University of Twente

research@linths.com

Abstract

Current automatic speech recognition (ASR) systems are greatly impacted by expressive speech, causing higher Word Error Rates (WER). Producing a large-scale training corpus with human expressive speech is a very laborious task. Very similar to data augmentation, we explore the field of expressiveness within a text-to-speech (TTS) system, creating a larger amount of speech data. Our speech synthesizer, ExpressTTS, aims to separately explore prosodic factors (pitch, energy, duration) and spectral tilt in a regularized latent space, while conditioning on the text and speaker. This way, we find expressive patterns that are natural in these contexts. Our non-autoregressive model parallelizes inference, allowing us to generate a large-scale corpus. We focus on the TTS part of augmentation pipeline. The qualitative evaluation shows that diverse but natural prosodic variations are found, but clear emotions are not audible. We see that our model generalizes consistently better to unseen in-domain utterances than the baseline. Quantitative analysis shows that the stability of the baseline and models is strongly influenced by the composition of the training corpus. There is a lack of expressive training data for our system, and padding it with neutral speech data yields domain mismatch which also inhibits model and baseline stability.

Nonetheless, the model generates speech with prosodic variation, and we find our model ExpressTTS consistently generalizes better to unseen in-domain data than the baseline GlowTTS. The user study suggests our model produces more diverse expressiveness than the baseline. To add, it creates significantly more intense emotion and style than the baseline. We conclude with directions on how to use our model for exploring expressiveness.

Index Terms: speech recognition, speech synthesis, data augmentation, prosody, computational paralinguistics

1. Introduction

1.1. Motivation

Modern automatic speech recognition (ASR) systems can transcribe speech audio for different conditions, such as speakers and acoustic environments. An ideal speech recognizer can also correctly transcribe speech with any form of human expressiveness. Current systems are however greatly impacted by expressive speech, causing higher Word Error Rates (WER) [2, 36]. Theoretically and proven empirically, training on expressive speech will mitigate this [36]. Popular corpus domains are commonly audiobooks [5, 16], which are not expressive in nature. Attaining a large annotated speech dataset in itself is a very laborious task, and forcing a certain level of expressiveness complicates speech collection more. Without expressive training data, ASR systems are not robust to expressive input – suffering increases in Word Error Rate (WER). After training on audiobooks, using distressed speech instead of neutral speech can increase the WER by 20-30% [2]. Different vocal effort levels, from whispering to shouting, can also increase a neutral baseline WER by 10-60% [36]. This can be mitigated by training specifically on only the same vocal effort level, decreasing WER by 5-40%. Alternatively, training on synthetic speech instead can also improve ASR systems [22].

Hypothesis We see limitations in expressive speech recognition and we hypothesize that including a wider or selected range of expressiveness in the training data may yield WER improvements – even if it is synthetic. Building a synthetic corpus through a text-to-speech (TTS) system allows for large-scale generation while controlling various conditions, such as text input and speaker selection.

Approach We propose a method to synthesize speech with diverse patterns of expressiveness for the same input. We call it ExpressTTS, and we discuss its use as a data augmentation model for ASR. The scope of this research is limited to the TTS part of the data augmentation pipeline. Further components of the pipeline – such as evaluating the WER improvement as a consequence of the resulting synthetic dataset – will be discussed as future work in Section 9.

1.2. Challenges

Building an expressive speech synthesizer comes with several challenges. Firstly, the model needs to learn from a small expressive corpus. Synthesizing expressive speech addresses the lack of expressive speech corpora. However, we need expressive speech to construct such synthetic corpus. As a consequence, the model needs to learn from a small expressive corpus. Too little training data overfit the model, making it unstable and causing poor or inconsistent performance on unseen utterances. We make a distinction between two types of unseen utterances: *lexically in-domain* and *lexically out-of-domain*. The model may not generalize to synthesize unseen texts from the corpus. They are lexically in-domain, i.e. from the same domain as the training texts. It is expected the model performs worse for texts from outside the corpus, and thus a different lexical domain.

Secondly, human expressiveness does not have an objective definition. This inhibits us from evaluating or training with a single direct metric or loss. Moreover, generative models are hard to evaluate – for the same reasons. There is not a single objective metric that indicates the generated sample is correct.

1.3. Research aims

The research aims to answer the following questions.

RQ1 How perceptually diverse in terms of expressiveness can we synthesize speech for a given text and speaker, while

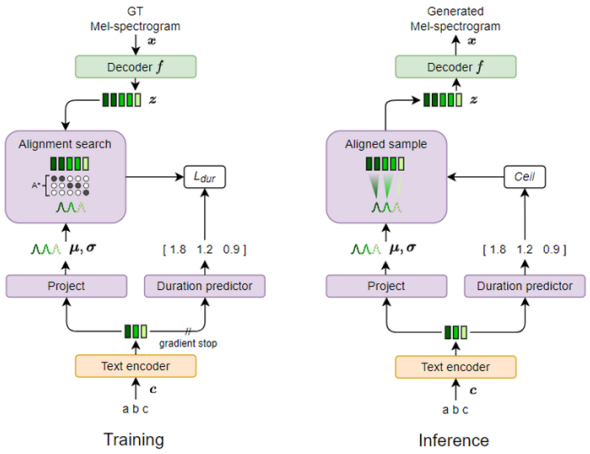


Figure 1: `base`. *The original Glow-TTS model [7]. It includes a reversible decoder with normalizing flows and a monotonic alignment search to match speech units with mel-spectrogram frames.*

maintaining a similar standard of naturalness across samples?

RQ2 How can a speech synthesizer construct and exploit a latent representation to generate perceptually diverse expressive speech for a given text and speaker?

RQ3 To what extent can the synthesizer increase diversity across samples, and at what cost in terms of computing resources?

Moreover, we note several requirements and desired properties for the system.

Firstly, the synthesizer is required to perform quick inference. Around 500 to 1000 hours is a common size for a multi-speaker ASR corpus [16]. As we are incorporating expressive patterns as well, we require a larger dataset. With limited resources for a data augmentation pipeline in mind, we aim at an inference speed that can comfortably produce around 3000 hours. For real-time TTS, the inference frequency should be high, surpassing 16 kHz. For our application, inference should be several orders of magnitude quicker. For example, if we want to build the 3000-hour corpus of 16 kHz speech within three days, the inference rate should be 666 kHz. Note that this does not include the computation time needed for training the speech synthesizer. Quick training for TTS will be greatly beneficial for performing various experiments, changing datasets and tweaking hyperparameters. This flexibility of changing datasets is also useful for real-life applications, where demands can change or 'better' expressive datasets may be released. Admittedly, human expressiveness is a subjective topic, so there is no objectively 'best' dataset to capture it. This very fact also invites experimentation with multiple datasets.

Secondly, the two common properties for TTS is intelligibility and naturalness [27]. While our evaluation focuses on the diversity of expressiveness, it includes a check for these properties to confirm basic TTS functionality.

1.4. Contributions

Our main contribution is the model ExpressTTS. It is designed to produce a large ASR corpus with high diversity efficiently. It explores expressiveness via a separate latent space, which is built from an expressiveness encoder. This way, expressiveness is conditioned on the input text and speaker. A secondary loss teaches this latent space to match the ground-truth energy, pitch, and spectral tilt contours. The model is an extension of Glow-TTS [7], featuring novel methods from Glow-TTS. It uses flow-based decoders, transformer encoders, and a monotonic alignment search. Contrary to most works, we do not use an encoder to represent the expressive ground truths. Instead, we use a reversed flow-based decoder. Lastly, we designed a new component for ExpressTTS: a latent merge, which merges the expressiveness latent space with another latent space.

Secondly, we analyze a multitude of speech corpora that could be used for expressive TTS. Our findings can help decision-making in corpora for similar research.

Thirdly, we provide several evaluation methods for expressive synthesizers (for ASR augmentation). We define an entanglement metric that measures entanglement between the two latent spaces. Moreover, the user study setup we designed can be used for similar systems – especially useful for benchmarking in this new field. We have customized the webMUSHRA [23] questionnaire source code to include these changes, and we have released it as open-source contributions[1].

Lastly, the nature of our work – exploring expressiveness using novel methods – shows ExpressTTS is a pioneering step in synthetic expressive ASR augmentation. One can use it to implement similar methods with additional restrictions when exploring expressiveness.

1.5. Outline

We provide background information about expressiveness and we define several key terms for expressive TTS in Section 2. We outline the state-of-the-art in speech synthesis, ASR augmentation, and style modeling in Section 3. Section 4 provides a detailed overview of the speech corpora we considered and choose. It provides important details, such as domain mismatch in our corpus, that are useful for interpreting our results. We describe our method in Section 5: the features, our method for exploring expressiveness, and the model ExpressTTS. Our experimental setup in Section 6 shows our baselines and defines our qualitative and quantitative evaluation methods. They include prosodic variation plots, several metrics, and a user study. Section 7 shows our results, which lead to the answering of the research questions in Section 8. Limitations and future work (Section 9) of the research are discussed as well. We end with conclusions in Section 10.

2. Background

2.1. Prosody

A speech recording contains information about the acoustic environment and about the speech uttered. The speech properties can be divided into phonemic and non-phonemic. To model human expressiveness, we focus on the non-phonemic properties only. To limit our scope, we will not consider non-phonemic vocalizations, e.g. laughing or sighing, but focus on *prosody*. Prosody refers to the non-phonemic properties of speech on a suprasegmental level, i.e. for syllables or larger units. There is no consensus on what all the attributes of prosody are. However, the generally agreed upon attributes are (i) the variation of pitch, (ii) variation of loudness, and (iii) durations of speech

[1] https://github.com/Linths/webMUSHRA-expressiveness

Category	Subcategories (non-exhaustive)	Effect	Has neutral
Emotion	Amusement, Anger, Disgust, Sleepiness, Sadness, Fear	Global	Yes
Interpersonal attitude	Authority, Contempt, Politeness, Irritation, Seduction	Global	Yes
Propositional attitude	Incredulity, Sarcasm, Surprise, Rhetoricity, Doubt, Confirmation, Obviousness	Local	Yes
Topical emphasis*	Beginning, Middle, End	Local	Yes
Style	Whispering, Shouting, Instructional, Broadcasting	Global	Yes
Marked tonicity*	Different interpretations of syntactically ambiguous utterances through tonicity	Local	No
Syntactic phrasing*	Different interpretations of syntactically ambiguous utterances through pauses	Local	No

Table 1: *Speech classes with prosodic effect from [29] with small adjustments. Terms marked with an asterisk are further explained in Appendix A.*

units [18]. With these, one can find patterns such as intonation, stress, tempo, and rhythm. Speakers use these patterns to convey additional information: to highlight speech units for focus and contrast. Moreover, physical correlations have been found between emotion and speech patterns, showing effects on pitch, timing, and voice quality (timbre) [18]. Specific emotions induce arousal of the sympathetic or the parasympathetic nervous system, causing changes in the heart rate, blood pressure, and salivation levels, which then influence speech loudness, pitch, and speaking rate. By measuring prosody to explore expressiveness, we in fact model both the speaker's emotional state and the additional information they want to convey.

A recent study defines several speech classes with prosodic effect [29] that carry such additional information. Table 1 shows an overview. The classes convey emotion, attitude, or clarification for a syntactically ambiguous utterance. This way, they classify speech samples into a non-exhaustive list of expressive categories and subcategories that are perceivable for listeners. Appendix A clarifies the terms topical emphasis, marked tonicity, and syntactic phrasing.

2.2. Definitions

We provide definitions for several key terms. As defined in [27], a speech sample is considered *natural* if it sounds like human speech. It is a perceptual phenomenon. To measure this, human listeners are often asked to rate the naturalness of the sample. This is commonly done with a Mean Opinion Score on a scale of 1 to 5: from bad to excellent. MUSHRA tests are also used, which employ a scale of 1 to 100 and require several extra test conditions, such as a reference sample [30]. Preference tests can also be used to ask listeners which sample is perceived as more natural.

Another perceptual phenomenon is expressiveness. We provide our own definition: a speech sample is considered *expressive* if it does not sound neutral, i.e. not encoding additional information besides the text uttered. The same techniques for measuring naturalness can be employed for expressiveness. For example, a preference test for expressiveness is performed in [10]. However, because listeners may have different interpretations of which audio deviations qualify as expressiveness, our tests provide expressive examples, providing an upper bound to anchor the MOS or MUSHRA scale.

Lastly, we define diversity in expressiveness. A set of samples has *diverse* expressiveness if the samples are perceived to convey different types of additional information, such as different emotions or attitudes. Similarly to expressiveness, we can evaluate it with human perception through a MOS, MUSHRA, or preference test.

3. Related work

3.1. Speech synthesis

Text-to-speech (TTS) systems can synthesize speech from any given text. They infer tremendous amounts of information to be able to convert from the low dimension of text to the significantly higher dimension of a waveform. Modern TTS systems are usually generative Encoder-Decoder models that focus on transforming text to a Mel-spectrogram, an intermediate representation of the waveform [34, 25, 15, 14, 19, 9, 7]. Then, a *vocoder* converts the Mel-spectrogram into a waveform.

To create correct and natural speech, several recent models use variational inference [14, 19, 7]. With a regularized latent space to represent any speech sample, the model obtains a well-structured organization of the speech data. This makes the model robust to outliers and unseen data, avoiding overfitting and generating diverse but natural speech samples. In Glow-TTS the latent space is as long as the speech sample, therefore it can generate meaningful speech for text of any length [7].

Another consideration for improving speech quality is how to model the dependencies between the text tokens and output tokens. Several methods use attention to capture long-term dependencies [34, 25, 19, 9, 7]. For specifically the dependencies between the output frames, some models use autoregression [34, 25, 15, 14, 19, 9]. While autoregressive models have been state-of-the-art for several years, they have a few disadvantages. As the spectrogram frame generation in an autoregressive model depends on the previously generated frame, the inference is slow. To add, it biases exploration as the autoregressive error accumulates, making the model less robust. Several autoregressive models arose that can parallelize inference using inverse autoregressive flows, resulting in sample inference at over 500 kHz [14, 19]. Their training is however complex, as it requires alignment with a pre-trained speech synthesizer.

The recent model Glow-TTS can model long-term dependencies without autoregression without compromising on output quality [7]. This not only improves inference speed but also robustness of the model. Figure 1 shows the system architecture. Glow-TTS' inference is 15.7 times faster than Tacotron 2 on average. Non-autoregression normally comes at the cost of training simplicity, as the output is predicted at once instead of frame by frame. Glow-TTS mitigates the training complexity by calculating the exact data likelihood in an efficient manner. During the forward pass, the spectrogram decoder uses normalizing flows to retrieve the exact representation of the output in the latent space. This latent space instance can be matched directly with the latent space distribution as determined by the input text, through a monotonic alignment search. Though, to ensure monotonicity, Glow-TTS requires Grapheme-to-Phoneme

conversion of the text input.

3.2. ASR augmentation

Data augmentation has been proven useful to improve speech recognition, and it can be considered standard practice. Research shows that even simple audio transformations allow for more diverse modeling of acoustic environments [32, 17] or prosody [6]. Noise is commonly added to clean signals. For example, real or simulated room impulse responses to add reverberation can give a relative WER improvement (WER-RI) of 45% . Inspired by cutout techniques in image augmentation, SpecAugment applies time warping, time masking, and frequency masking to spectrograms, resulting in up to 46% WER-RI [17]. Other augmentation methods for acoustics are low-pass smoothing or additive Gaussian noise. Augmentation policy search, first popularized in computer vision, can be used to maximize utility of such augmentation methods, especially when enforcing consistency measures [32].

Prosody augmentation can model change in speaker aspects. To make the prosody of adult speech childlike, [6] use vocal-tract length normalization, explicit pitch modification, and duration modification, resulting in 32% WER-RI on child speech. To convert emotive utterances to a neutral version, [21] apply uniform and non-uniform modification of pitch, duration, and energy. The modification factors for the three emotive categories anger, happiness, and compassion followed from dataset analysis. WER improved by an absolute 5% across all three categories.

Instead of augmented real speech, one can use augmented synthetic speech to improve ASR. Such speech can be created by forcing enhancements in the speech synthesis process. For example, [22] apply data augmentation with TTS in two fields. They model speaker diversity by synthesizing speech with speakers sampled from existing embeddings. Randomly generated speaker embeddings proved less beneficial. To add lexical diversity, they feed the TTS system new texts sampled from a trained language model. WER improvements are seen for both augmentation experiments, but do depend on (i) the augmentation ratio of the training data – synthetic versus original data – and (ii) the size of the original dataset. The larger dataset benefited less from augmentation because it likely contains more diversity on itself. Moreover, they concluded that an in-domain synthesizer, trained on the same dataset as for testing the ASR, gives better WERs than an out-domain synthesizer. Their experimental setup can be used as a follow-up for our work. Then, one can explore the augmentation ratio, and consider the effects of domain adaptation when using datasets of different domains and sizes. Their speaker-domain findings also suggest that a random generation of expressiveness will be less useful than sampling. Lastly, we use multi-speaker TTS which they deemed necessary for ASR augmentation.

3.3. Style modeling

Modeling style in speech synthesis can be done in various ways. Firstly, one can take different definitions for style: prosody, certain prosodic features, or the audio variance unexplained by text and speaker factors. Secondly, the eventual purpose can motivate different modeling approaches. While synthesizers may model style to solely improve speech naturalness, we focus on more specific uses.

To create speech with the specific prosody of sadness, [4] use a simple prosody transfer method. They synthesize speech with equal-pitch and equal-duration syllables, and then apply voice transformation with pitch and duration factors copied from a reference speech sample. Other prosody transfer methods avoided the need for labeled prosody by using unsupervised representations of prosody within the TTS system. With a reference encoder, they create a global prosody embedding from a reference spectrogram or spectrum [26, 33, 37]. This embedding will then be used in the TTS decoder [26, 37] or TTS encoder [33]. There might be certain constraints to transfer prosody. In [26], prosody could only be transferred if the reference text (nearly) matches the target text. Furthermore, when the reference speaker differs from the target speaker, the transfer results show apparent traces from the reference speaker's voice. This indicates entanglement of speaker and style.

Another prosody transfer work with style embeddings, called Global Style Tokens (GSTs), use it for prosody control as well [34]. Because they enforced a maximum of ten unique GST embeddings, the resulting GSTs would model commonly found patterns. Experiments show that a single GST influences the whole speech contours for pitch, intensity, speaking rate, and 'emotion' (quoted). They also found that scaling the GSTs with positive or negative factors can intensify or lessen the style effect, respectively – though this may lead to unintelligible speech.

The design of speech synthesizer Glow-TTS allows for certain prosody control [7]. With generative flows, it can directly convert a spectrogram from a sample $\boldsymbol{z}$ of a Gaussian latent space with a learnt mean $\boldsymbol{\mu}$ and unit variance σ. If we however use temperature T as variance, we retrieve $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} * \boldsymbol{T}$, where $\boldsymbol{\epsilon}$ is a standard normal distribution sample. It was found that by only varying $\boldsymbol{\epsilon}$, the resulting speech samples showed different intonation patterns. Moreover, controlling only the temperature $\boldsymbol{T}$ showed control over the speech pitch. Another crucial component of Glow-TTS is the duration predictor, which learns from observing the text-speech alignment search during training, to then impose durations onto the phonemes during inference. Imposing a manually chosen duration vector allows for direct control of the speaking rate.

Style can also be captured to make the TTS system robust to noise and highly biased data. A hierarchical Variational Auto-Encoder (VAE) was used to improve stability and fidelity when working with non-studio speech data [22]. The hierarchical VAE creates one global style embedding from a reference spectrogram, much like the aforementioned prosody transfer methods, and additionally creates a local embedding for each spectrogram slice. Then, each TTS decoder step takes into account the global embedding and the relevant local embedding. Including the hierarchical VAE gives a WER improvement of 1%.

4. Dataset

Our work addresses a lack of large expressive speech corpora, but to build our pipeline, we still require an expressive corpus. The dataset chosen will heavily influence the way our speech synthesizer learns and how the ASR is evaluated. Table 2 contains most of the corpora considered. All contain American English speech. We found five possible multi-speaker datasets that contain at least 9.1 hours of transcribed and highly expressive speech. Three of them are developed for multimodal emotion recognition: MELD [20], IEMOCAP [3], and CMU-MOSEI [35]. The other two are built for emotion control or transfer in speech synthesis: EmoV-DB [1] and ESD [38]. All five corpora contain emotional labels and the (quite uniform) emotion distribution can be carefully sampled. Emotion labels are not necessary for our pipeline because they can be left out as input

Corpus	Purpose	#Speakers	#Emotion labels	#Utterances, distinct	#Utterances, non-distinct	Size (h)	Avg. speaker size (h)
EmoV-DB [1]	Emotion control	4	5	593	6.9 K	9.5	2.4
ESD [38]	Emotion transfer	10	5	350	17.5 K	10.3	1.0
MOSEI [35]	Emotion recognition	1000	6	23.5 K	23.5 K	66.0	0.1
IEMOCAP [3]	Emotion recognition	10	10	10.0 K	10.0 K	11.5	1.1
MELD [20]	Emotion recognition	6	6	13.0 K	13.0 K	13.5	1.9[2]
Blizzard [8]	TTS	1	/	40.6 K	40.6 K	41.0	41.0
LJSpeech [5]	TTS	1	/	13.1 K	13.1 K	24.0	24.0

Table 2: *Training corpora considered. Our experiments use EmoV-DB and LJSpeech.*

features. However, they can provide insight during our system analysis.

For any of the emotion recognition datasets, a first inspection suggests good audio quality. However, for TTS, speech quality needs to be very high with clean acoustic environments. MELD, which contains fragments from the TV sitcom Friends, has noise ranging from subtle background sounds to an overlapping laugh track [20]. IEMOCAP, with 12 hours of speech from scripted and spontaneous dialogues from 10 actors, has speaker overlap [3]. The CMU-MOSEI dataset contains 65 hours of at least 1000 speakers from different YouTube videos [35]. The dataset is a strongly mixed source which may lead to domain mismatch in a lexical and acoustic sense. A more consistent dataset is preferred to facilitate the learning of expressive patterns. Besides that, the speech quality was found to be lacking, as well as the transcriptions [24]. However, while CMU-MOSEI has clear disadvantages, it can still be interesting to run a side experiment on it as the high size and number of speakers may uncover a high range of expressive patterns.

Rather than emotion recognition, EmoV-DB and ESD are made for synthesis purposes. Our data analysis also confirmed they both meet TTS quality standards. EmoV-DB is created for emotional expressiveness control in voice generation systems [1]. It contains 9.5 hours of scripted speech in 5 emotive categories, voiced by 2 female and 2 male speakers. The emotions are acted out in an exaggerated manner with non-verbal vocalizations such as laughter and yawning. ESD is made to improve emotional style transfer in voice conversion. It contains 10.3 hours over 5 female and 5 male speakers, using 5 emotions. A very important note is that they both use *parallel utterances*. Utterances are purposely re-recorded in different conditions: each speaker and emotion. Arguably, this encourages disentanglement of emotional patterns and the text input, as any text input is seen in various styles. There is a clear downside, however. The corpus size of around 10 hours is already considerably small for TTS, but the number of distinct utterances is only in the several hundreds. A lack of lexical diversity inhibits the speech synthesizer from properly training for the complex task of TTS. We confirmed this through experimentation with both corpora. Moreover, in our case, low lexical diversity may inhibit the generalization of the emotional patterns found.

To ensure our model can in fact perform speech synthesis, we pad a highly expressive corpus with a common large-scale TTS corpus. As highlighted in [22], this does lead to *domain mismatch*. Lexically and acoustically, the two corpora have different distributions. This can also inhibit convergence of training and decrease model stability. For the highly expressive corpus, we chose EmoV-DB over ESD based on the number of distinct utterances. As the neutral corpus we use LJSpeech, a single-speaker 24.0 hour corpus of audiobooks. As can be seen in Table 2, we also retrieve an imbalance in utterances and hours per speaker. The LJSpeech speaker has a tenfold of hours over one EmoV-DB speaker. EmoV-DB uses the emotions {*Amused, Angry, Disgusted, Neutral, Sleepy*}. When merging the corpora, we assign the label *Neutral* to all LJSpeech samples – creating an emotion imbalance as well. These imbalances also decrease model stability.

While we performed our experiments on the aforementioned composite corpus, we listed one other corpus. With the stability issues in mind, we see potential in the Blizzard 2013 corpus [8] as well. This is a large-scale single-speaker TTS corpus. It contains animated and emotive storytelling. As it is a regular TTS corpus, there are no emotion labels. An additional experiment would be to train our synthesizer on just the Blizzard corpus, trading off some expressiveness for a high increase in model stability.

[2]The MELD subset includes six main speakers and a multitude of non-frequent speakers which make up around 1/7th of the corpus.

5. Method

5.1. Approach

We designed ExpressTTS, an expressive speech synthesizer, to produce a large corpus with diverse expressiveness efficiently. Based on the high inference speed and quality, we use Glow-TTS [7] as a base system. Moreover, Glow-TTS already offers the possibility to explore diversity within speech, providing a baseline for ExpressTTS. In Glow-TTS, spherical Gaussian distributions represent the spectrograms. Thus at inference time one can trade off between naturalness – sampling near the mean – and diversity – sampling away from the mean.

The strength of style embeddings in many works motivates our choice to create a separate latent space for expressiveness. This way, we can model variation in expressiveness separately from any other type of variation, such as pronunciation. For a more fine-grained representation, we use local embeddings which are on a frame level. We use a well-structured regularized representation that motivates the generation of diverse but natural samples. For this, we use spherical Gaussian distributions for expressiveness embeddings, like Glow-TTS has for spectral embeddings. This expressiveness latent space offers the same trade-off between naturalness and diversity. Section 5.3 describes our method to systematically sample diverse variations within our model ExpressTTS and the baseline Glow-TTS.

To model human expressiveness, we focus on the non-phonemic properties only. To limit our scope, we will not consider non-phonemic vocalizations, e.g. laughing or sighing. In-

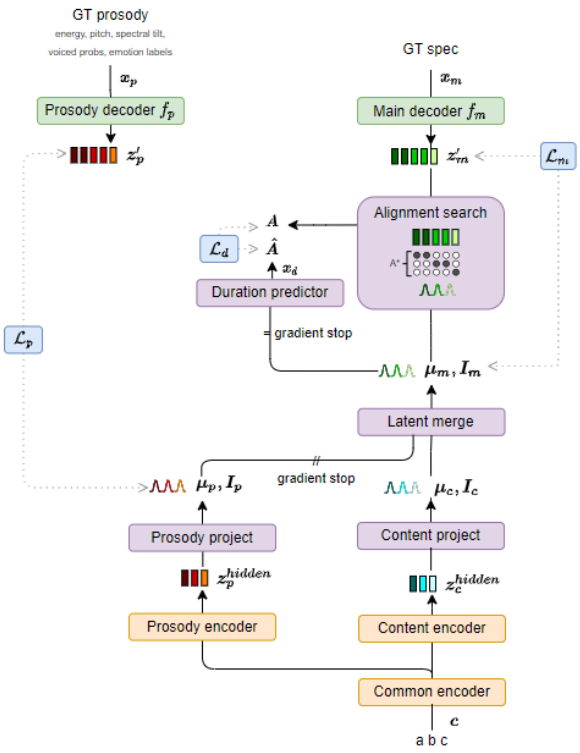


Figure 2: *ExpressTTS, version 1.4. It extends the Glow-TTS [7] base model with reversible decoders for prosodic features, a prosody encoder, a prosody loss, and a latent merge. At inference time, it samples from the prosody latent space to synthesize speech with diverse expressiveness.*

stead, we focus on prosody. Section 5.2 describes the exact features we use. Prosodic properties are specific for the text uttered and the speaker's voice. Certain languages use for example lexical stress to distinguish semantics. A speaker might have a more high-pitched voice or a low speaking rate in general. To fully explore the spectrum of human expressiveness, we condition the prosody on the text and speaker. More specifically, we control the text and speaker to model expressiveness in different conditions. Prior works in prosody modeling however show these two factors are prone to entanglement with expressiveness, so our system evaluation also focuses on the disentanglement of them.

Figure 2 is a detailed overview of our model, ExpressTTS. We made the following additions to the original Glow-TTS model: (i) an energy-pitch decoder, (ii) a prosody encoder, and (iii) a merge for the prosody latent space and the content latent space. These changes include an additional loss, which minimizes mismatch between the prosody predicted from the input text and speaker and the prosody extracted from the ground-truth sound wave. In addition, the model still learns from the main loss (which minimizes spectral mismatch) and the duration loss (which minimizes phone-frame alignment mismatch).

After a description of what ground-truths we use to build the expressiveness latent space from (Section 5.2), and how we sample from it (Section 5.3), we show how our model integrates this latent space in Section 5.4 and later.

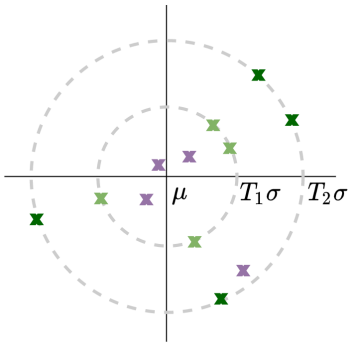


Figure 3: *Sampling methods used, visualized in a 2D example. Purple depicts samples drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. Both shades of green depict samples drawn by taking random angles at a fixed distance $T\boldsymbol{\sigma}$ from $\boldsymbol{\mu}$.*

5.2. Features

To recognize and emulate patterns of expressiveness, we focus on several features: *pitch*, *energy*, *spectral tilt*, and the *emotion labels*. Pitch is decomposed into the *voiced probabilities* – the probability a frame is voiced – and the interpolated pitch. Spectral tilt can be considered a composite factor that considers pitch and energy. The interpolated pitch has less sudden drops than the non-interpolated pitch. Instead, these drops are modeled by the voiced probabilities. The emotion labels are added to encourage the clustering of expressive patterns per emotion. While only pitch and energy concern prosody, for the sake of simplicity we refer to all of the aforementioned features as the *prosodic features*.

As a first normalization step, we apply a logarithm to the pitch and energy values found. The voiced probabilities and spectral slope naturally fall in a small and better-distributed range of values. A one-hot encoding is used for the five emotion labels. Z-normalisation is applied to all features, except for the emotion labels. Per feature, the mean and standard deviation needed for the normalization are calculated from the training subset.

5.3. Exploration of expressiveness

Our model and the baseline model contain Gaussian latent spaces, representing spectral or prosodic variation. We can use different methods to sample from a Gaussian latent space. The most straightforward method, as used in Glow-TTS for regular speech synthesis, is to sample $\hat{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, or $\hat{\boldsymbol{z}} = \boldsymbol{\mu} + \epsilon * \boldsymbol{\sigma}$. Per definition, this yields natural patterns. Their limited experimentation with sampling lead them to sample $\hat{\boldsymbol{z}} = \boldsymbol{\mu} + \epsilon * T\boldsymbol{\sigma}$, while varying only the temperature T or only the noise ϵ.

To force diverse samples in a systematic way, we fix the distance to μ – the most likely sample – to be $T\sigma$ with a T of choice. This creates a sphere of possible samples. As shown in the 2D example in Figure 3, the last unknown is the angle. We calculate these samples as follows.

$$\hat{\boldsymbol{z}} = \boldsymbol{\mu} + T * \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \tag{1}$$

Here, $\boldsymbol{a}$ is the reference vector that has the chosen angle. One can sample it from any Gaussian with zero mean. The variance does not matter; we divide $\boldsymbol{a}$ by its ℓ^2-norm to bring the coordinate on the unit circle – yielding unit distance to $\boldsymbol{\mu}$.

We designed the following methods to choose $\boldsymbol{a}$.

- **Random angles.** We randomly draw $\boldsymbol{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- **Independent angles.** To ensure the samples represent independent information, we can force every $\boldsymbol{a}$ vector to be orthogonal to the previous samples.
- **Dimension extremes.** Every sample represents one extreme of one latent dimension. Note this is a specific case of the previous category. With N as the number of dimensions, we use the following equations to attain $2N$ samples. $\forall i \in \{1, ..., N\}$:

$$\boldsymbol{a}_i^{pos} = (0_1, .., 0_{i-1}, 1_i, 0_{i+1}, .., 0_N) \tag{2}$$

$$\boldsymbol{a}_i^{neg} = (0_1, .., 0_{i-1}, -1_i, 0_{i+1}, .., 0_N) \tag{3}$$

To quickly find the influence of input changes in the latent space, the dimension extremes are the preferred reference vectors – given that N is reasonably small enough. Otherwise, the random angles are a good alternative to still explore.

5.4. Encoders

Prosody encoder The prosody encoder learns the prosody latent space distribution $\boldsymbol{\mu_p}, \boldsymbol{\sigma_p}$ given the text for any speaker. As it is a distribution, it can not only represent the most likely prosody for a given text and any speaker, but also less likely possibilities for prosody. It allows modeling human expressiveness as a trade-off between naturalness (i.e. likelihood) and diversity. The backpropagation from the prosody loss $\mathcal{L}_p$ ensures the prosody encoder includes various expressive patterns. The prosody encoder does not backpropagate via the main loss $\mathcal{L}_m$ to ensure it does not represent non-prosodic information.

Content encoder The text encoder learns $\boldsymbol{\mu_c}, \boldsymbol{\sigma_c}$, the latent space distribution of the text and speaker. Using the main loss $\mathcal{L}_m$, the variation within the spectrogram is explained by the prosody encoder and the content encoder together. As the prosody encoder is taught to solely represent prosodic information, the remaining information – e.g. phoneme pronunciation – should be provided by the content encoder. Because the ground-truth spectrogram contains prosody as well, it is possible prosodic information leaks into the text encoder via the main loss. This would be redundant, as the prosody encoder already represents prosody.

Common encoder The prosody encoder and text encoder share a common encoder, to detect fundamental patterns in the given text. The common encoder learns from both the prosody loss and the main loss. We build the common encoder from 4 layers, and the follow-up encoder from 2 layers. Encoders for speech synthesis commonly have around 6 layers.

5.5. Decoders & predictor

Main decoder The main decoder operates on a frame-level. When reversed during the forward pass, the main decoder gives $\boldsymbol{z'_m} = f_m^{-1}(\boldsymbol{x}_m)$. This is the exact representation of the ground-truth spectrogram $\boldsymbol{x_m}$ in the main latent space $\boldsymbol{Z_m}$. The decoder reversal is possible through invertible functions in its normalizing flows. No adaptations were made to this Glow-TTS component.

Prosody decoder The internal design of the prosody decoder is hugely similar to the main decoder. It operates on a frame-level. When reversed during the forward pass, the prosody decoder return $\boldsymbol{z'_p} = f_p^{-1}(\boldsymbol{x}_p)$. This is the exact representation of the ground-truth prosodic features $\boldsymbol{x_p}$ in the prosody latent space $\boldsymbol{Z_p}$.

Duration predictor The duration predictor is the same as the one in Glow-TTS. It operates on a phone-level and predicts the duration for each phone. We feed it the main latent space sample.

5.6. Latent merge

The latent merge learns to merge the prosody latent space distribution $\boldsymbol{\mu_p}, \boldsymbol{\sigma_p}$ with the content latent space distribution $\boldsymbol{\mu_c}, \boldsymbol{\sigma_c}$, such that it forms the main latent space $\boldsymbol{\mu_m}, \boldsymbol{\sigma_m}$ with as little mismatch with the decoded ground-truth spectrogram as possible. This is learnt via the main loss $\mathcal{L}_m$. This is a crucial component in the model. More information about the implementation details will be reported in a later version of this document.

5.7. Losses

Main loss The main loss minimizes the mismatch between $\boldsymbol{z}'_m$: the latent space instance that exactly represents the spectrogram, and $\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m$: the latent space distribution representing the text and speaker information. The main latent space distribution is built by merging $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p$ and $\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c$, the latent space distributions of the prosody encoding and content encoding. We calculate the main loss $\mathcal{L}_m$ with the negative log-likelihood of the spectrogram x_m given the context $\boldsymbol{c}$ (i.e. text and speaker). The main decoder $f_m : \boldsymbol{x}_m \rightarrow \boldsymbol{z}'_m$ is reversible via normalizing flows. Therefore, we can use the *change of variables* to calculate the log-likelihood of $\boldsymbol{z}_m$ instead – which is $\boldsymbol{z}'_m$ after alignment. The change of variables is the second term in Equation 5.

$$\mathcal{L}_m = -\log P_{\boldsymbol{X}_m}(\boldsymbol{x}_m|\boldsymbol{c}) \tag{4}$$

$$= -\log P_{\boldsymbol{Z}_m}(\boldsymbol{z}'_m|\boldsymbol{c}) - \log \left| \det \frac{\partial f_m^{-1}(\boldsymbol{x}_m)}{\partial \boldsymbol{x}_m} \right| \tag{5}$$

The main latent space $\boldsymbol{Z}_m$ is a Gaussian distribution with mean $\boldsymbol{\mu}_m$ and standard deviation $\boldsymbol{\Sigma}_m = \boldsymbol{I}_m$. We fix the standard deviation to be this constant, as done in the original experiments of Glow-TTS. Similarly to Glow-TTS, we parameterize the data and prior distributions with network parameters $\boldsymbol{\theta}$ and the alignment function $\boldsymbol{A}$. If $A(j) = i$, then $\boldsymbol{z}'_{m,j}$ (the j-th frame of $\boldsymbol{z}'_m$) is distributed with mean $\boldsymbol{\mu}_{m,i}$ and variance $\boldsymbol{\sigma}_{m,i}$ (the elements of $\boldsymbol{\mu}_m$ and $\boldsymbol{\sigma}_m$ that refer to the i-th phone). We denote the number of mel-spectrogram frames by T_{mel}.

$$\log P_{\boldsymbol{Z}_m}(\boldsymbol{z}'_m|\boldsymbol{c}; \boldsymbol{\theta}, A) = \sum_{j=1}^{T_{mel}} \log \mathcal{N}(\boldsymbol{z}'_{m,j}; \boldsymbol{\mu}_{m,A(j)}, \boldsymbol{\sigma}_{m,A(j)}) \tag{6}$$

$$\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m = \text{latent\_merge}(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c) \tag{7}$$

Prosody loss The prosody loss minimizes the mismatch between $\boldsymbol{z}_p$: the latent space instance that exactly represents the target prosody, and $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p$: the latent space distribution representing the prosody implied by the text and speaker embeddings. Here the variance is fixed too, as $\boldsymbol{\sigma}_p = \boldsymbol{I}_p$. Because the prosody decoder $f_p : \boldsymbol{x}_p \rightarrow \boldsymbol{z}_p$ is reversible via normalizing

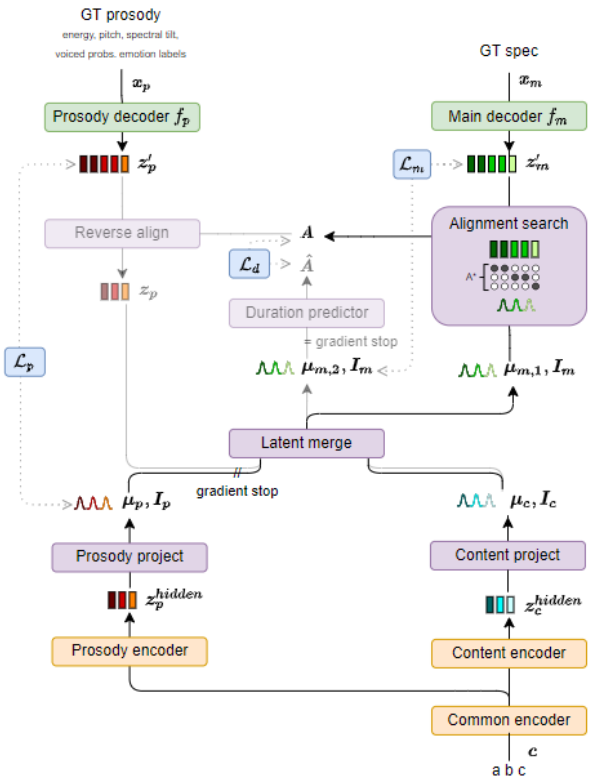


Figure 4: *ExpressTTS, version* `1.5`.

flows, we can calculate $\mathcal{L}_p$ with the change of variables and a simple Gaussian likelihood, similarly to $\mathcal{L}_m$.

$$\mathcal{L}_p = -\log P_{\boldsymbol{X}_p}(\boldsymbol{x}_p|\boldsymbol{c}) \tag{8}$$

$$= -\log P_{\boldsymbol{Z}_p}(\boldsymbol{z}_p|\boldsymbol{c}) - \log \left|\det \frac{\partial f_p^{-1}(\boldsymbol{x}_p)}{\partial \boldsymbol{x}_p}\right| \tag{9}$$

$$\log P_{\boldsymbol{Z}_p}(\boldsymbol{z}'_p|\boldsymbol{c};\boldsymbol{\theta},A) = \sum_{j=1}^{T_{mel}} \log \mathcal{N}(\boldsymbol{z'}_{\boldsymbol{p},\boldsymbol{j}};\boldsymbol{\mu}_{p,A(j)},\boldsymbol{\sigma}_{p,A(j)}) \tag{10}$$

Duration loss The duration predictor f_{dur} is trained with a mean square error. It uses a gradient stop sg to ensure the loss will not propagate to the rest of the architecture. In the following equation, x_d denotes the 'ground-truth' durations as found by the alignment search.

$$\mathcal{L}_d = MSE(f_{dur}(sg[\boldsymbol{z}_m]), \boldsymbol{x}_d) \tag{11}$$

5.8. Model variations

`1.4` and `1.5` The model as explained is called version `1.4`. It has one drawback that may severely limit to what extent exploring the $\boldsymbol{Z}_p$ can cause prosodic changes in the spectrogram. During the training forward pass, the latent merge gets fed $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_c$. This means that the resulting $\boldsymbol{\mu}_m$ is based on the most likely prosody for a given text and any speaker. It essentially loses important prosodic information. During the backward pass, the main loss may find significant mismatch for non-neutral speech. The spectrogram would then contain prosody clearly modeling a certain emotion, but $\boldsymbol{\mu}_m$ is based on the most likely prosody, which is most probably neutral. This unnecessary spectral mismatch will then be propagated to the latent merge and the content encoder – not the prosody encoder because of the deliberate gradient stop. As a consequence, the main latent space will learn to model prosodic variation instead of the prosody latent space.

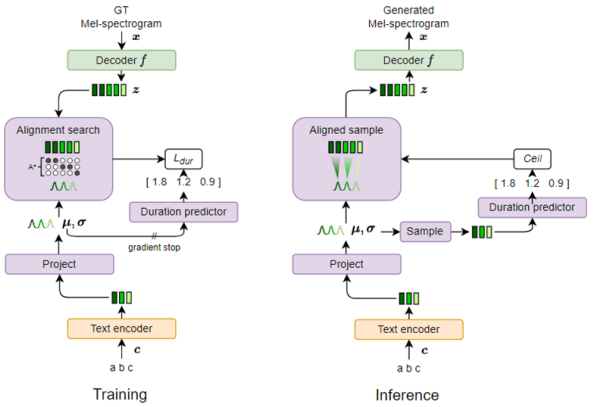


Figure 5: `base adapted`. *Adaptation of the Glow-TTS model [7], used as additional baseline. The change allows for prediction of variation in phone durations for one text and speaker.*

Figure 4 shows version `1.5`, which addresses this issue. In 1.5 the latent merge is fed $\boldsymbol{z}'_p$, the exact latent space representation of the prosodic ground-truth. The resulting $\boldsymbol{\mu}_m$ then incorporates this prosodic information, which in turn decreases the spectral mismatch. The latent merge will also learn to properly merge more diverse instances of $\boldsymbol{Z}_p$, which is useful for when we explore during inference time. Because the latent merge operates on a phone level, the prosodic ground-truth embedding $\boldsymbol{z}'_p$ needs to be reverse-aligned. An alignment needs to be calculated for this, which is why we added a prior forward pass. In the first pass, $\boldsymbol{\mu}_p$ is used to attain $\boldsymbol{\mu}_m$, which gives the alignment. In the second pass, the reverse alignment returns $\boldsymbol{z}_p$, which is together with $\boldsymbol{\mu}_c$ fed to the latent merge. The second pass is visualized as greyed-out blocks in Figure 4. The following steps are then standard procedure.

`1.4 switch` and `1.5 freeze` Around two thirds of our corpus consists of neutral speech. To encourage generation of highly expressive speech despite the imbalance, we adjust our training methods. In `1.4 switch`, we freeze the prosody decoder for LJSpeech batches. To ensure the prosody encoder can still generate patterns for any text input, we do not freeze it for LJSpeech batches. Instead, the prosody encoder switches from learning from the prosody loss to the main loss. For version `1.5` the latter is not possible, which is why we then freeze both the prosody decoder and encoder.

6. Experimental setup

Model configuration We train for 5000 epochs and use a batch size of 128; 32 per GPU. We use an Adam optimizer with the Noam learning rate schedule. The other details are in Ap-

pendix E.

Baselines Figures 1 and 5 depict our baselines. As a first baseline, we train the Glow-TTS model on the EmoV-DB and LJSpeech composite corpus. This `base` model predicts spectral variations through the main latent space. However, the duration predictor only gives a single prediction for the phone durations – meaning it does not model variation in durations. ExpressTTS does, so to provide a baseline with equal influence on the output audio we created another baseline. By feeding the duration predictor samples from the latent space (Figure 5), the `base adapted` model can predict diverse phone durations for a given utterance and speaker.

6.1. Analysis overview

In the next subsections, we define the methods which we use to answer the research questions one by one. To provide an overview, we outline which methods are relevant for each research question.

RQ1 Prosodic variation plots, individual listening test, main loss, and quantitative listening test

RQ2 Prosodic variation plots, latent space inspection, prosody loss, duration loss, entanglement score, and cluster score

RQ3 Quantitative listening test

6.2. Qualitative analysis

To answer the research questions, we not only analyze the model output but also the model components by performing ablation studies. By analyzing the constructed prosodic latent space, we can see how and which expressive patterns are incorporated. While the quantitative analysis enables us to make more generalizable and statistically justified conclusions, qualitative analysis plays an important role in generative systems such as TTS.

Prosodic variation plots We analyze the influence of diverse samples from the prosody latent space while keeping the content latent space sample fixed. Then, we can plot the *predicted prosodic features*, attained by simply decoding the prosody latent space samples. Figure 6 is an example. Next to this, we can plot the *extracted prosodic features* from the resulting audio wave. This waveform is vocoded from the spectrogram, which is the result of decoding the main latent space sample – constructed from the prosody sample and the fixed content sample. The extracted prosodic features help answer **RQ1** as it visualizes diversity in prosody. The predicted prosodic features help answer **RQ2** as it is a direct visualization of what prosodic information the model stores in the latent space – regardless of how well-preserved it is in the latent merge and the following steps in the main branch of the system.

The same plotting technique can be applied to plot the prosodic influence of the main latent space. The possibilities are limited to only plotting the *extracted prosodic features* from the waveform. Inspecting the main latent space in ExpressTTS garners understanding of entanglement between the prosodic latent space and the content latent space – answering **RQ2**. Further explanation about evaluating entanglement follows in section 6.3.

Individual listening test To investigate the *perceptual* diversity and naturalness as mentioned in **RQ1**, a human ear is needed. Listening to samples confirms whether the prosodic diversity results in an audible change in audio and in expressiveness. We first listen for these properties ourselves without a strict question setup to get an indication. Note that this qualitative method is mostly used as a sanity check and guide to shape further experiments.

Latent space inspection We can inspect how well-structured the latent space is by plotting the locations of samples in the latent space. Dimensionality reduction techniques such t-SNE and UMAP [31, 11] allow us to visualize the latent space. Visually inspecting the clusters and relative distances allows us to hypothesize which aspects are modeled or disregarded. This all helps us answer **RQ2**. Note that this qualitative method is mostly used as a sanity check and guide to shape further experiments.

6.3. Quantitative analysis

Several of the aforementioned qualitative techniques can be executed on a larger scale. This can not only give more data-driven conclusions but also help answer **RQ3** as varying sampling parameters sheds light on the efficiency of the system.

Losses The prosody loss helps answer **RQ2** as it is a direct indicator of the mismatch between the predicted prosody and the ground-truth prosody – showing the capabilities of $\boldsymbol{Z_p}$. Similarly, the main loss helps answer **RQ1** as it is a direct indicator of spectral mismatch, and thus indirectly prosodic mismatch. The capabilities of the main latent space directly affect the resulting speech and its prosodic aspects. The duration loss helps answer **RQ2** as the predicted phone durations are based on the samples of the prosodic latent space.

Entanglement score To generate speech samples that are only diverse in prosody, and consistent in other (spectral) aspects, it is important to have a disentanglement between the prosodic latent space and the content latent space. We measure how entangled $\boldsymbol{Z_p}$ and $\boldsymbol{Z_c}$ are by calculating the mutual information between these Gaussian distributions. In this calculation, we use Kullback-Leibler divergence, where the main latent space $\boldsymbol{Z_m}$ is taken as the joint distribution. Both marginal distributions are spherical Gaussians, allowing us to significantly simplify the calculation.

$$s_{ent} = I(\boldsymbol{Z_p}; \boldsymbol{Z_c}) \tag{12}$$

$$= D_{KL}(\boldsymbol{Z_m} \| \boldsymbol{Z_p} \boldsymbol{Z_c}) \tag{13}$$

$$= (\boldsymbol{\mu_m} - \boldsymbol{\mu_p})^2 + (\boldsymbol{\mu_m} - \boldsymbol{\mu_c})^2 \tag{14}$$

The entanglement score helps us answer **RQ2** as it sheds light on the information the prosodic latent space captures.

Clustering score To find how well-clustered embeddings of different emotions are in the prosodic latent space, we use a clustering score. A high score indicates better clustering. For this, we measure the intra-cluster distance and the inter-cluster

distance. We denote the number of emotions by E, and the prosodic latent space embeddings by $\boldsymbol{x}$.

$$s_{cluster} = \frac{d_{intracluster}(Z_p)}{d_{intercluster}(Z_p)} = \frac{\frac{1}{E}\sum_{i=1}^{E}\frac{1}{N_i}\sum_{j=1}^{N_i}(\boldsymbol{x_{i,j}} - \boldsymbol{\mu_i})^2}{\frac{1}{E}\sum_{i=1}^{E}(\boldsymbol{\mu_i} - \boldsymbol{\mu})^2} \tag{15}$$

In the variance calculation, we keep the $\boldsymbol{Z}_p$ dimensions to see how well-clustered an individual dimension is. The cluster score shows how well-structured the prosodic latent space is, answering **RQ2**.

Quantitative listening test User tests give a subjective evaluation of the expressiveness and naturalness of the synthetic speech. In the quantitative listening test, we find direct answers to **RQ1** and **RQ3**. We draw samples, using the same method, from the latent spaces of our model and baselines. This way, we can compare which system results in higher diversity in expressiveness. By varying the temperature in the sampling methods, we also answer how much diversity is gained by sampling more, and how it affects the naturalness. Then, we can assess how many samples need to be drawn, in which configurations, to reach certain levels of diversity. This way, we can directly show the trade-off between the cost of resources and the overall diversity. The rest of this section details the motivation and question design of our listening test.

The listening test targets the following core questions.

1. How perceptually diverse in terms of expressiveness and how natural are the synthesized samples from ExpressTTS and GlowTTS for a given text and speaker when taking samples across the three most prominent prosody latent space dimensions?
2. Does adding new samples, attained by increasing the distance from earlier samples to the latent space mean, influence the overall naturalness and diversity of expressiveness for the whole set of samples?

We determine which versions we use for the ExpressTTS model and the baseline based on prior observations from our other evaluation methods. In a similar way, we determine which dimensions are the most prominent, by finding which dimensions cluster emotions the best and which dimensions create the most prosodic variations according to the variation plots.

The first major component is naturalness. For speech synthesis, naturalness is commonly measured with a mean opinion score (MOS) or a MUSHRA [30] test, which is then compared with the baseline score of human speech. We use MUSHRA, the MUlti Stimulus test with Hidden Reference and Anchor. It is developed to evaluate multiple audio stimuli with accurate and reliable results with a panel of only 20 listeners. Users rate the property, for example, naturalness, on a scale from 0 to 100. Besides multiple test samples, a MUSHRA page includes a reference sample, a hidden version of this reference, and two anchors. An example can be seen in Figure 13 in Appendix D. The anchors are used to scale the user's ratings. For the lower anchor, one chooses a sample that should consistently receive a low score. For the upper anchor holds the inverse. For naturalness, we use a human speech sample for both the upper anchor and the reference. Ideally, this receives a score of 100. No lower anchor is used as there is no robust candidate for it [12]. The reference gives the user a sample to compare the test stimuli with. The hidden reference is used to exclude listeners from the analysis if they provide poor ratings for it. After post-screening the listeners, we carry out the step-by-step statistical analysis guide from the MUSHRA specification [30]. It includes exploratory data analysis and ANOVA.

For the second component, diversity in expressiveness, we adapt the standard MUSHRA test. Our adaptations are usable for similar research questions. Our goal is to measure expressiveness as perceivable and meaningful variations from neutral speech. One cannot simply ask listeners to "rate the expressiveness" of speech samples; this leaves a lot of room for interpretation for the listeners – making results less reliable and the assessment procedure for listeners more tiresome. Instead, assessors classify samples as speech classes with prosodic effect and rate the intensity of the effects. Each category is treated separately in a MUSHRA screen. We use all speech classes from Table 1, except for interpersonal attitude as [29] found it has a recognition rate of under 50% for female speakers. For each category, the user will choose a fitting subcategory, if any effect is present. If they perceive no effect, they rate the intensity 0. If they find an unspecified subcategory is more fitting, they can add their own. For marked tonicity and syntactic phrasing, we provide two possible interpretations instead of subcategories. We use for both categories specific utterances that are syntactically ambiguous but are clarified through prosody.

With this method, we see for every sample which types of expressiveness listeners recognize. We determine the diversity of expressiveness by counting distinct subcategories present in the groups of samples for every model, using a cut-off threshold for intensity. Multiple thresholds can be used, such as 30 (poor), 50 (fair), 70 (good), and 90 (excellent) [30]. This way, our systems can be compared in a more fine-grained manner.

The thoroughness of MUSHRA tests yields high statistical significance, but it comes at the cost of assessment time. Therefore we only evaluate two systems: one ExpressTTS version and one baseline. We also limit sampling to three dimensions for this reason. We evaluate the effects of scaling samples by using two sampling temperatures. We evaluate seven categories: naturalness and six types of expressiveness. Including hidden references and anchors, these parameters result in a MUSHRA test of 14 screens with 14 sliders. According to [28] the estimated assessment time is 20 minutes when split the workload over two groups. While additional time is needed for choosing subcategories, less time is necessary for the expressiveness intensity sliders as the majority of sliders are expected to be on 0. Samples are expected to be neutral for most categories, showing at most a few types of expressiveness simultaneously.

7. Results

Training time The models have been trained for 5000 epochs. Training took any version of ExpressTTS around 10 days on 4 NVIDIA A100 GPUs. The baselines both took 6 days in the same setting. The runtime complexity for inference in bulk will be reported as well.

7.1. Qualitative analysis

Prosodic variation plots During the training of the models, we visualized the prosodic variations predicted for several fixed training utterances. Figure 6 shows the output for model `1.4 switch`, exploring the extremes of the 9 $\boldsymbol{Z_p}$ dimensions. Note that sample 0 is the most likely prosody, $\boldsymbol{\mu_p}$. The ground-truth sample shows the extracted prosodic values from the recording

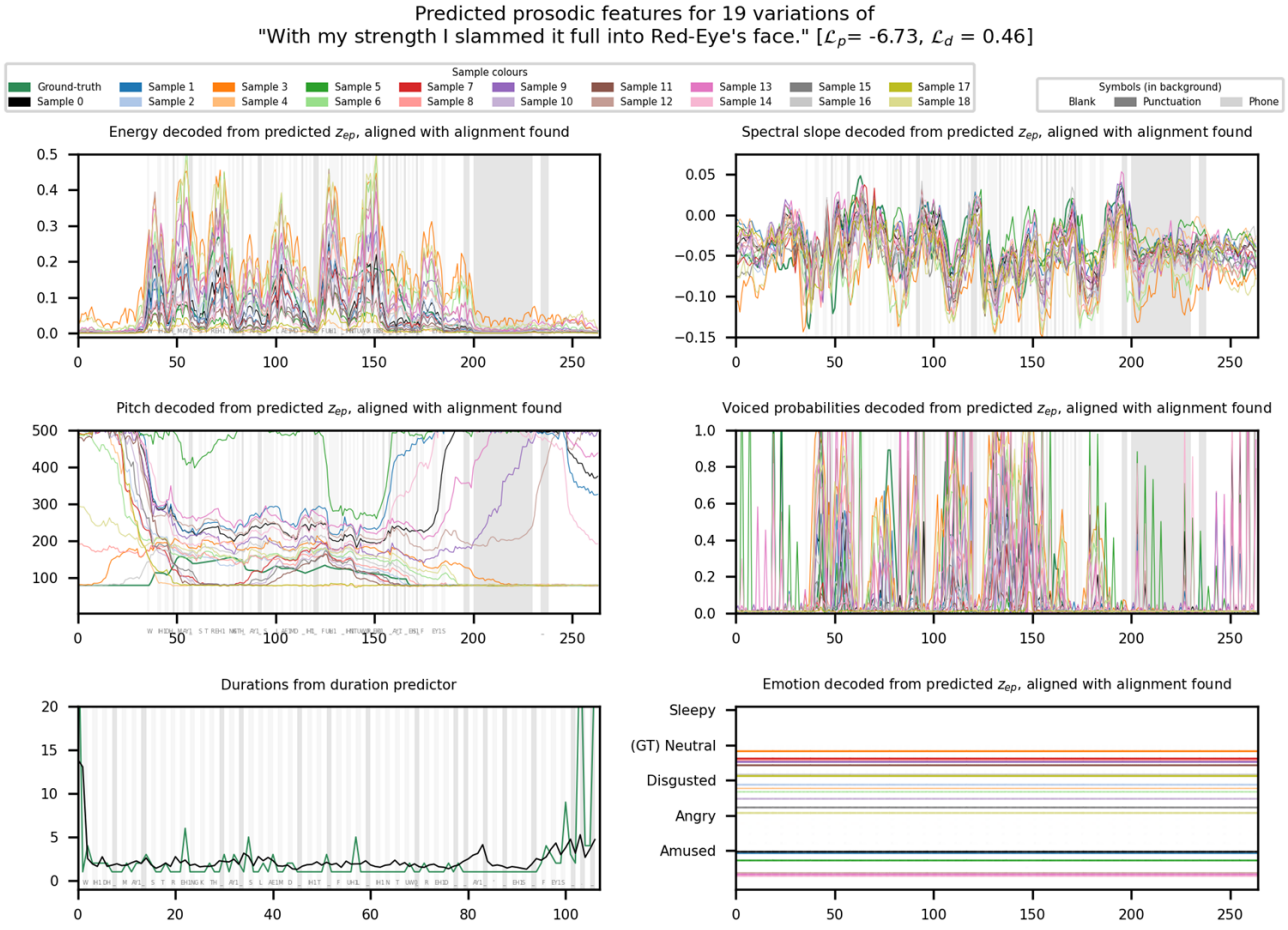


Figure 6: *Predicted posodic variations for one EmoV-DB training utterance and an EmoV-DB speaker. Used model* `1.4 switch` *at 5000 epochs. Sample 0 is the most likely prosody,* $\boldsymbol{\mu_p}$*. Any other samples represent extremes of the* $\boldsymbol{Z_p}$ *dimensions at temperature* $T = 2$*.*

uttered by the correct speaker neutrally. This does not mean our model samples should match exactly; here we aim to model other emotions too. Besides, even with the same conditions – text, speaker, and emotion – the speech output can still take various forms.

We see diverse prosodic patterns arise, which mostly come across as natural. For energy, the sampling seems to mainly adjust the scale. The spectral slope seems to be both scaled and shifted. In the interpolated pitch, there is an obvious shift between samples, but we also see other patterns arise. Across samples, we see a difference in the pitch range, the shapes, and the slopes. The voiced probabilities mostly change in scale only; just several of the samples seem to add peaks. The emotion plot suggests various emotions are predicted.

The phone durations vary per sample too, but it is not included in this plot. All the aforementioned features have been adjusted with the same alignment for this visualization.

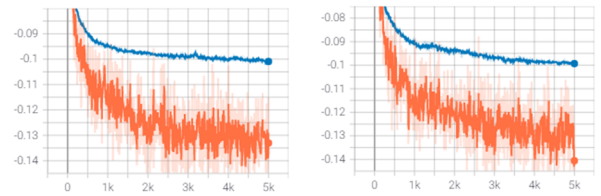
Similar plots for other models are in Appendix B. For them, we see similar effects.

Individual listening tests We use the following results to shape initial ideas about the systems, and to guide other experiments such as the user test.

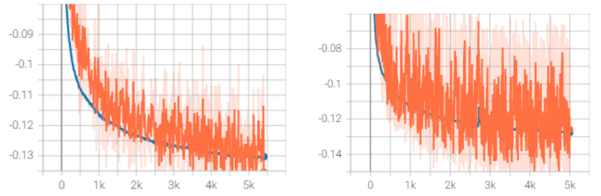
We use one experiment with the `base` system on LJSpeech only. It shows very natural output as the most likely output, but also a few samples can be found with distinct yet natural expressiveness. However, many samples had to be explored to find these; around 80 samples were taken at random angles with a temperature of around 4.

The other experiments are performed in a systematic manner and on the composite EmoV-DB and LJSpeech corpus. Per model and latent space, we cherrypicked 2 perceptually diverse samples from 20 samples – attained by sampling the latent space at random angles at distance $T\boldsymbol{\sigma}$ from $\boldsymbol{\mu}$. The most likely sample $\boldsymbol{\mu}$ is included as well. The means from the baselines already show the composite corpus gives less natural and smooth output. Exploring $\boldsymbol{Z_m}$ yields variation in pitch, energy, overall intonation, and more. Some samples even seem to model a slightly different speaker. This is possible because the main latent space controls all spectral information instead of only prosody.

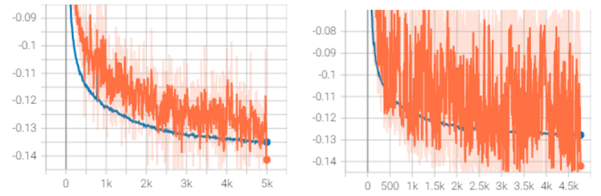
Of all models, version `1.4` has the most natural-sounding $\boldsymbol{\mu}$ sample. It sounds at least as natural as the baselines' $\boldsymbol{\mu}$ samples. Our other model versions have considerably worse naturalness – meaning it could not train properly enough for just the



(a) `base` (b) `base adapted`

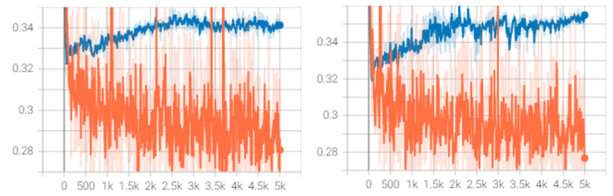


(c) `1.4` (d) `1.4 switch`

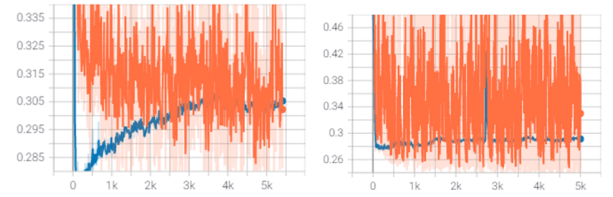


(e) `1.5` (f) `1.5 freeze`

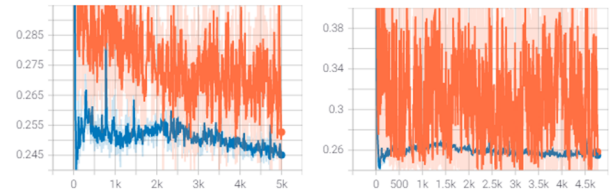
Figure 7: *Main loss throughout epochs per model. Orange depicts the training subset, blue the validation subset. A smoothing factor of 0.9 is used.*



(a) `base` (b) `base adapted`



(c) `1.4` (d) `1.4 switch`



(e) `1.5` (f) `1.5 freeze`

Figure 8: *Duration loss throughout epochs per model. Orange depicts the training subset, blue the validation subset. A smoothing factor of 0.9 is used.*

TTS task itself.

Using the composite corpus, none of the samples in any configuration seems to convey a specific emotion. For all our model versions we can however hear prosodic changes when exploring $\boldsymbol{Z_p}$. Among those are whispering, less clear voiced probabilities (i.e. slurring), or sharp changes in energy – which may come across as somewhat aggressive.

When exploring $\boldsymbol{Z_m}$ instead, we can still find prosodic variations. In versions `1.5` and `1.5 switch`, these main latent samples do not come across as very natural. For versions `1.4` and `1.4 switch`, on the other hand, these samples sound as natural prosodic variation – something that we would like to see modeled by $\boldsymbol{Z_p}$ instead.

Latent space inspection We constructed latent spaces by reverse decoding samples from the EmoV-DB dataset via the prosody decoder. Inspection shows emotions are reasonably clustered. An example can be found in Appendix C. Admittedly, emotion labels are part of the prosodic features, meaning this information is already explicitly supplied to the prosody decoder.

7.2. Quantitative analysis

While training, we calculated the model losses and the symmetric entanglement score. As the baselines do not have a prosodic latent space, there is no prosody loss plot, entanglement plot, or cluster score for them.

Losses For all losses and the entanglement score of any system, we can see instability. All time series concerning the training subset show a wide range of values and sudden switches. All plots in fact depict the smoothed[3] time-series. The grayed-out lines depict the unsmoothed series. It is important to note this instability is seen for the baselines as well – admittedly in a lighter form though. Earlier experimentation showed that similar plots for the `base` system trained on LJSpeech only do not show this instability.

In the following analyses, we focus on the validation set. Figure 7 shows the main loss converges around -0.10 for both baselines, while all model variants converge around -0.13. Model `1.5` even gets as low as -0.13, meaning it best minimizes spectral mismatch.

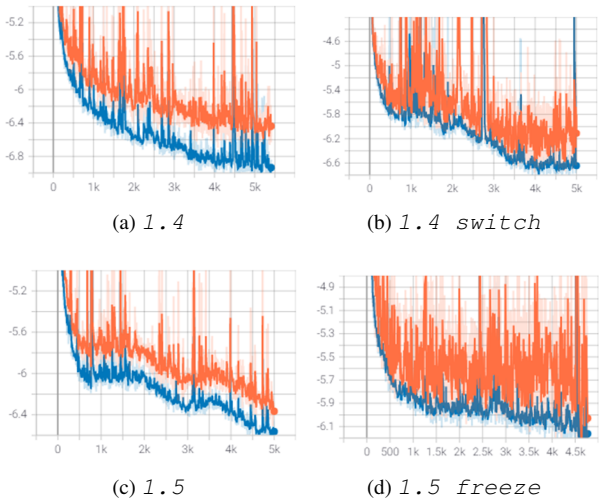
In Figure 8, we see the duration loss converges around 0.34 and 0.36 for the `base` and `base adapted` respectively. The models `1.4`, `1.4 switch` and `1.5 freeze` converged around 0.30, 0.28 and 0.26 respectively. Interestingly, version `1.5` does not seem converged yet and reports the lowest value found in the higher epochs: it went consistently below 0.25.

As shown in Figure 9, the `1.5` and `1.5 freeze` prosody losses do not seem converged. Versions `1.4` and `1.4 switch` arguably settled around -7.0 and -6.6. Versions `1.5` and `1.5 freeze` reached -6.6 and -6.2 respectively, but they might still decrease considerably with more epochs.

[3]The smoothing uses an exponential moving average with a factor of 0.9

Model	Prosodic latent space dimensions								
`1.4`	0.924	0.396	0.662	1.577	3.300	0.347	0.426	0.204	0.253
`1.4 switch`	1.906	0.343	0.175	4.158	0.770	0.143	0.208	0.971	0.214
`1.5`	0.382	1.220	1.224	0.967	4.152	0.350	0.407	0.811	0.222
`1.5 freeze`	1.719	0.449	0.576	0.531	1.649	0.242	1.154	0.558	0.308

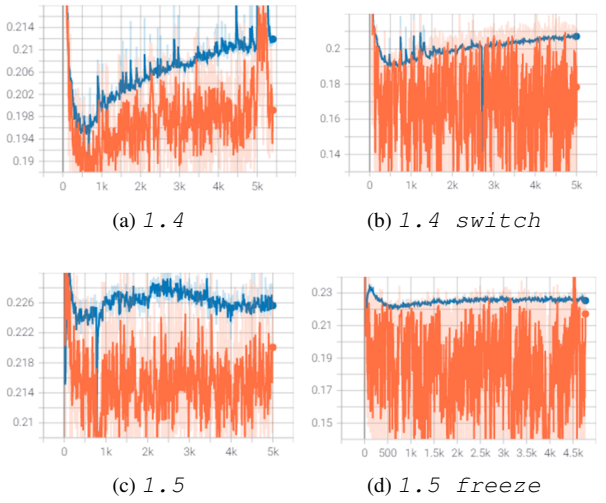
Table 3: *Cluster score for the prosodic latent space of every model. It measures emotion clustering per dimension. Values under 1 indicate low clustering: the variance within clusters is larger than the variance between clusters. Values over 1 indicate the inverse.*



(a) *1.4* (b) *1.4 switch*

(c) *1.5* (d) *1.5 freeze*

Figure 9: *Prosody loss throughout epochs per model. Orange depicts the training subset, blue the validation subset. A smoothing factor of 0.9 is used.*



(a) *1.4* (b) *1.4 switch*

(c) *1.5* (d) *1.5 freeze*

Figure 10: *Entanglement score s_{ent} throughout epochs per model. Orange depicts the training subset, blue the validation subset. A smoothing factor of 0.9 is used.*

Entanglement score Figure 10 depicts the entanglement scores. Versions `1.4 switch` and `1.5 freeze` seem to have converged around 0.21 and 0.23 respectively. Both validation set time-series are more stable than its `1.4` and `1.5` counterparts. In version `1.4`, entanglement seems to grow higher than 0.21, but the curvature could suggest total convergence within several thousand epochs. For `1.5` the future path is not easily predictable. It does seem it will stay around 0.22. The most disentangled version is `1.4 switch`. Moreover, versions `1.4` and `1.5` both have a clear dip before reaching 1000 epochs. This may be attributed to the fact the initial prosodic and content latent space did not contain strong patterns necessary for TTS yet, so they might have been modeling very distinct patterns at first.

Cluster score Table 3 contains the cluster scores for the prosodic latent spaces of our models. We see that for every latent space most dimensions show low emotion clustering. However, `1.4`, `1.4 switch`, and `1.5` all have at least one dimension scoring over 3, clustering emotions well. Their prosodic latent spaces thus retain emotional information. This can also be claimed, but less strongly, about version `1.5 freeze`.

Quantitative listening test Following our observations, we use version `1.4` as the model. It shows the highest stability in the quantitative analysis and the most natural-sounding samples in the individual listening test. We focus on two $\boldsymbol{Z}_p$ dimensions: the fourth and the fifth. They are the best clustered, and they show interesting variations in the qualitative prosodic variation plot in Figure 11a in Appendix B. As a baseline, we use `base`. Both baselines score similarly in the quantitative and qualitative analysis, but `base` has a slightly better duration loss. Calculating the cluster score for $\boldsymbol{Z}_m$ of `base` lead us to select the 20th and 30th dimensions. For both model and base, we determine T_1 by reverse decoding the emotional samples and finding the average distance to the latent space mean. As a second temperature, we use $T_2 = 3T_1$. We use $T_1 = 0.75$ and $T_2 = 2.25$ for the model, and $T_1 = 0.5$ and $T_2 = 1.5$ for the base. We refer to the subset of samples taken at temperatures $\{0, T_1, T_2\}$ as 'all', and the samples taken temperatures $\{0, T_1\}$ as 'small'.

We use two groups for the user study, each rating half of the categories. Naturalness, interpersonal attitude, propositional attitude, and marked tonicity are rated by 18 anonymous assessors and the remaining categories by 16 assessors. Results are collected via our customized webMUSHRA [23] questionnaire.

ANOVA analysis, Tukey HSD post-hoc tests, and independent samples T-tests are used to test several hypotheses. In Table 4, we see the model sounds less natural than the base at

Category	System	Mean	Model vs Base Mean Diff.
Naturalness*	Base	45.61	**-9.047**
	Model	36.57	
	High anchor	75.89	
	Reference	97.79	
Emotion	Base	16.77	5.188
	Model	21.96	
	Low anchor	55.25	
	Reference	59.50	
Interpersonal attitude	Base	12.47	-0.281
	Model	12.19	
	Low anchor	32.05	
	High anchor	32.97	
	Reference	39.89	
Propositional attitude	Base	12.54	-2.211
	Model	10.33	
	Low anchor	15.11	
	High anchor	50.03	
	Reference	56.84	
Topical emphasis	Base	20.72	**-7.521**
	Model	13.20	
	High anchor	79.84	
	Reference	82.53	
Style	Base	17.74	5.347
	Model	23.08	
	Low anchor	54.25	
	Reference	83.72	
Marked tonicity	Base	34.70	**-9.462**
	Model	25.23	
	High anchor	50.79	
	Reference	52.63	
Syntactic phrasing	Base	38.15	-3.965
	Model	34.19	
	High anchor	96.19	
	Reference	95.84	

Table 4: *Intensity scores of the user study. For naturalness, instead of intensity, naturalness is rated. T-tests determine whether model has higher scores than base. Values in bold are significant at the 0.05 level. Table 8 in Appendix D contains the full table.*

a statistically significant level. Though not statistically significant overall, we see that the model might provide worse expressive intensities than the base, except for emotion and style. The model has significantly less intense topical emphasis and marked tonicity than the base. Moreover, we notice that all of the anchors and references consistently score higher than the base and model; even the low anchor does.

As for expressive diversity, we count the distinct subcategories found which are rated with an intensity of at least 1 out of 100. Table 5 shows that the model may have more diverse expressiveness than the base – except for topical emphasis, marked tonicity, and syntactic phrasing. None of the T-tests were statistically significant at the 0.05 level.

Table 6 shows that sampling at an additional higher temperature, i.e. *scaling up*, yields a significant increase in naturalness for the base, and a significant decrease for the model. Table 5 shows that scaling up may diversify expressiveness. Only in one setting – the category emotion with the model – we see that this diversification is significant.

Category	System	Mean	All vs Small Mean Diff.	Model vs Base Mean Diff.
Emotion	Base	3.56	0.438	0.500
	Model	4.06	**1.188**	
Interpersonal attitude	Base	3.11	0.368	0.056
	Model	3.17	0.778	
Propositional attitude	Base	3.11	0.889	0.444
	Model	3.56	0.889	
Topical emphasis	Base	3.06	0.438	-0.187
	Model	2.88	0.438	
Style	Base	3.69	0.750	0.063
	Model	3.75	0.875	
Marked tonicity	Base	1.89	0.167	0.000
	Model	1.89	0.000	
Syntactic phrasing	Base	1.81	0.188	-1.250
	Model	1.69	0.125	

Table 5: *Expressive diversity results of the user study, i.e. the count of unique subcategories a user annotated across all samples for a system. Firstly, T-tests determined whether sampling at another temperature increases the subcategory diversity – listed in the 'All vs Small' column. A second sequence of T-tests determined whether the model has more unique subcategories than the base. Values in bold are significant at the 0.05 level. Table 7 in Appendix D contains the full table.*

System	Subset	N	Mean	All vs Small Mean Diff.	Sig.
Base	All	171	45.61	**5.656**	0.046
	Small	95	39.96		
Model	All	171	36.57	**-6.18**	0.023
	Small	95	42.75		

Table 6: *Naturalness scores of the user study at different subset levels. Two T-tests determine whether sampling from two temperatures ('all') increases naturalness as opposed to sampling from only the lower temperature ('small'). Values in bold are significant at the 0.05 level.*

8. Discussion

We found the following answers to the research questions.

Firstly, to answer **RQ1**, our findings suggest our model ExpressTTS in all variations and the baselines can all produce prosodic variations. The individual listening tests however show that the baselines can produce other variations than of the prosodic kind, for example affecting the voice of the speaker. We found that naturalness is compromised to produce variations for any of our systems. These subjective observations are to be confirmed and further quantified by the MUSHRA listening tests. Moreover, we see that not the complete spectrum of spectral diversity is captured by any of our systems, as the main loss curve has been unstable for each of them. We compare it with the stable main loss for the base system trained on only LJSpeech, and we provide two non-exclusive explanations. The instability can signal difficulty to produce the full range of expressive diversity in the spectrograms. Additionally, the insta-

bility can simply be the result of two separate distributions produced by the two corpora EmoV-DB and LJSpeech. Nonetheless, ExpressTTS consistently generalizes better to unseen in-domain utterances than the baselines. The user test shows that in general the model shows higher expressive diversity than the baseline, except for categories that operate on a very high local level: topical emphasis, marked tonicity, and syntactic phrasing. However, none of these results were statistically significant. Moreover, the user study showed the model may have more intense emotions and style than the base. Emotion and style are both global effects. We conclude that for global expressive effects, our model can be preferred over the base.

Secondly, to answer **RQ2**, we find that our constructed latent spaces retain emotion information and that sampling from them produces prosodic variations. The prosodic latent space is extremely good at matching prosodic variation, reaching a negative log-likelihood of around -6.8 on validation data. Moreover, the prosodic latent space is nearly disentangled from the content latent space; the mutual information comes down to only 0.21 nats or 0.14 bits of information. The entanglement score also shows signs of instability, which can again be attributed to variation in the domain (EmoV-DB or LJSpeech) or expressiveness. Moreover, across all of the losses, metrics, and the listening test we see the instability grows more for our models when freezing component(s) for the LJSpeech samples.

Lastly, to answer **RQ3** we use the results from the user test. We found that adding samples of a higher temperature may increase the expressive diversity. While this cannot be stated with statistical significance for most settings, we do see a significant increase in emotion diversity when we use our model. However, scaling up pays a price: it causes a significant decrease in naturalness.

There are several limitations to our research. We do not have a method that measures prosodic variations in a quantified manner. Our prosodic variations plots are made per utterance, providing a limited perspective. Our user study, while quantified, does have several limitations. We only use one speaker of one gender, two dimensions, two temperatures, and two systems, meaning we cannot generalize the results to other speakers, dimensions, or systems. Moreover, our panel size was too small to reach statistically significant results for several categories. This is especially a concern with the expressive diversity test, where we use an aggregated value over all samples. Here, the number of data points is the number of users: 16 or 18. Moreover, several users noted that they experienced *listener fatigue* to some extent. This can influence the quality of the results filled in during the later parts of the questionnaire.

9. Future work

We suggest several adaptations and additions to our research.

Firstly, we strongly suggest attaining a more stable corpus, without domain mismatch. We hypothesize this can significantly change the results. To clarify, we found that for our model `1.4` some important prosodic variations were modeled in the wrong place, the main latent space, instead. This can be explained by an architectural choice which is handled in version `1.5`. As a consequence, version `1.5` relies more on the richness of the emotional corpus, which is too small to make an entirely stable model with. This architectural choice is a literal trade-off between expressive diversity and model stability, where the model stability issues forced us to evaluate with model `1.4`.

We provide further reasoning for this hypothesis. The base model on a large neutral corpus, LJSpeech [5], does suggest even emotional patterns can be captured, but not in an efficient manner. As a small side experiment we fully trained ExpressTTS on only the large-scale corpora LJSpeech, and the preliminary results sound promising. We find similar prosodic variation as with the base model on LJSpeech, with higher sampling efficiency. We suggest further experimentation with LJSpeech only or a slightly more expressive large-scale corpus such as Blizzard [8]. We do trade in expressiveness to a degree as these corpora are audiobooks.

Secondly, we suggest using the entanglement metric as a loss to encourage latent space detanglement. The entanglement metric can also be augmented with a asymmetric version which measures how well $\boldsymbol{Z}_c$ can predict $\boldsymbol{Z}_p$, and vice versa. This can be estimated by training two feed-forward networks.

Thirdly, configurations and hyperparameters can be largely varies. Important fields of focus are: prosodic latent space dimensionality, the prosody decoder architecture, the latent merge architecture. Specifically, we would simplify the prosody decoder, as it now is as complex as the main decoder which captures more detailed variations.

Fourthly, we could our approach for the prosodic latent space. We can add a global token, as a residual besides the local prosody embeddings. This may encourage learning global variations, such as emotion and style, separately from local variations, such as topical emphasis. Moreover, we can change our sampling method to vary on a local level, by using different angles and scales at different parts of the utterance. This way, we may sample more local variations.

As a fifth point, we see improvements for the user study. More listeners helps the statistical significance, and more groups to spread the workload decreases listener fatigue. The user study can be used to evaluate many more configurations, such as more speakers, utterances, sample angles, sample temperatures, systems, and latent spaces. It can also be used to measure more phenomena, such as prosody transfer. Moreover, nuance can be added to the expressive diversity analysis by considering expressive subcategories as only relevant from different intensity thresholds, such as 20, 40, or 60. Alternatively, we can implement a different metric for expressive diversity other than a count, which could incorporate the intensity ratings as weights.

As a sixth point, we refer to the quantitative prosodic variation plots we describe in Appendix F.

Lastly, it is important to directly evaluate the system with an ASR system. For this, one needs to finish the whole ASR pipeline: construct the whole corpus, train an ASR system and a baseline, and measure Word-Error-Rate to see if it decreases. One can then also vary the augmentation ratio of the constructed corpus: the ratio of synthetic speech versus human speech.

10. Conclusions

While the lack of highly expressive data and domain mismatch has led to unstable models and baseline, we find our model ExpressTTS consistently generalizes better to unseen in-domain data than the baseline GlowTTS. The best model, `1.4`, generates speech with prosodic variation. The user study suggests it produces more diverse expressiveness than the baseline. To add, it creates significantly more intense emotion and style than the baseline.

We conclude with directions to explore prosodic variations with our model `1.4`. Note that if one wants to model local ef-

fects, one may consider using the baseline GlowTTS instead. For global effects, especially emotion and style, we advise systematically sampling across the prosodic latent space dimensions. We advise the fourth and fifth dimensions specifically, and the temperatures 0.75 and 2.25. More temperatures can be added, but note that increasing the temperature will decrease naturalness. To more freely explore prosodic variations of any kind, we suggest sampling across more dimensions. When resources are not a concern, we suggest using many (random) angles to cover the prosodic latent space even better.

11. References

[1] Adaeze Adigwe et al. "The emotional voices database: Towards controlling the emotion dimension in voice generation systems". In: *arXiv preprint arXiv:1806.09514* (2018).

[2] Frédéric Aman, Véronique Aubergé, and Michel Vacher. "Influence of expressive speech on ASR performances: Application to elderly assistance in smart home". In: *International Conference on Text, Speech, and Dialogue*. Springer. 2016, pp. 522–530.

[3] Carlos Busso et al. "IEMOCAP: Interactive emotional dyadic motion capture database". In: *Language resources and evaluation* 42.4 (2008), pp. 335–359.

[4] Lyes Demri, Leila Falek, and Hocine Teffahi. "Contribution to the Design of an Expressive Speech Synthesis System for the Arabic Language". In: *International Conference on Speech and Computer*. Springer. 2015, pp. 178–185.

[5] Keith Ito and Linda Johnson. *The LJ Speech Dataset*. `https://keithito.com/LJ-Speech-Dataset/`. 2017.

[6] Virender Kadyan, Syed Shanawazuddin, and Amitoj Singh. "Developing children's speech recognition system for low resource Punjabi language". In: *Applied Acoustics* 178 (2021), p. 108002.

[7] Jaehyeon Kim et al. "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search". In: *arXiv preprint arXiv:2005.11129* (2020).

[8] Inc. Lessac Technologies. *Voice Factory audiobook recordings for Blizzard 2013*. `https://www.cstr.ed.ac.uk/projects/blizzard/2013/lessac_blizzard2013/`. Accessed: 2021-12-11.

[9] Naihan Li et al. "Neural speech synthesis with transformer network". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6706–6713.

[10] Rui Liu et al. "Expressive tts training with frame and style reconstruction loss". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2021).

[11] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[12] Thomas Merritt et al. "Comprehensive evaluation of statistical speech waveform synthesis". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 325–331.

[13] Devang S Ram Mohan et al. "Ctrl-P: Temporal Control of Prosodic Variation for Speech Synthesis". In: *arXiv preprint arXiv:2106.08352* (2021).

[14] Aaron van den Oord et al. "Parallel WaveNet: Fast High-Fidelity Speech Synthesis". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 3918–3926. URL: `http://proceedings.mlr.press/v80/oord18a.html`.

[15] Aaron van den Oord et al. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* (2016).

[16] Vassil Panayotov et al. "Librispeech: an ASR corpus based on public domain audio books". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2015, pp. 5206–5210.

[17] Daniel S Park et al. "Specaugment: A simple data augmentation method for automatic speech recognition". In: *arXiv preprint arXiv:1904.08779* (2019).

[18] Oudeyer Pierre-Yves. "The production and recognition of emotions in speech: features and algorithms". In: *International Journal of Human-Computer Studies* 59.1-2 (2003), pp. 157–183.

[19] Wei Ping, Kainan Peng, and Jitong Chen. "Clarinet: Parallel wave generation in end-to-end text-to-speech". In: *arXiv preprint arXiv:1807.07281* (2018).

[20] Soujanya Poria et al. "Meld: A multimodal multi-party dataset for emotion recognition in conversations". In: *arXiv preprint arXiv:1810.02508* (2018).

[21] VV Vidyadhara Raju et al. "Importance of non-uniform prosody modification for speech recognition in emotion conditions". In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2017, pp. 573–576.

[22] Andrew Rosenberg et al. "Speech recognition with augmented synthesized speech". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2019, pp. 996–1002.

[23] Michael Schoeffler et al. "webMUSHRA—A comprehensive framework for web-based listening tests". In: *Journal of Open Research Software* 6.1 (2018).

[24] Imran Sheikh et al. "Sentiment analysis using imperfect views from spoken language and acoustic modalities". In: *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. 2018, pp. 35–39.

[25] Jonathan Shen et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4779–4783.

[26] RJ Skerry-Ryan et al. "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron". In: *international conference on machine learning*. PMLR. 2018, pp. 4693–4702.

[27] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[28] The University of Edinburgh The Centre for Speech Technology Research. *Evaluation of speech synthesis researchers guide*. https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/Speak14To15/evaluation.pdf. 2014.

[29] Alexandra Torresquintero et al. "ADEPT: A Dataset for Evaluating Prosody Transfer". In: *arXiv preprint arXiv:2106.08321* (2021).

[30] International Telecommunication Union. *BS.1534 : Method for the subjective assessment of intermediate quality level of audio systems*. https://www.itu.int/rec/R-REC-BS.1534/en. 2015.

[31] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[32] Gary Wang et al. "SCADA: Stochastic, Consistent and Adversarial Data Augmentation to Improve ASR". In: *Proc. Interspeech 2020* (2020), pp. 2832–2836.

[33] Yuxuan Wang et al. "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5180–5189.

[34] Yuxuan Wang et al. "Tacotron: Towards end-to-end speech synthesis". In: *arXiv preprint arXiv:1703.10135* (2017).

[35] AmirAli Bagher Zadeh et al. "Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2236–2246.

[36] Petr Zelinka, Milan Sigmund, and Jiri Schimmel. "Impact of vocal effort variability on automatic speech recognition". In: *Speech Communication* 54.6 (2012), pp. 732–742.

[37] Guangyan Zhang et al. "Estimating Mutual Information in Prosody Representation for Emotional Prosody Transfer in Speech Synthesis". In: *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE. 2021, pp. 1–5.

[38] Kun Zhou et al. "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 920–924.

12. Appendix

A. Speech classes with prosodic effects

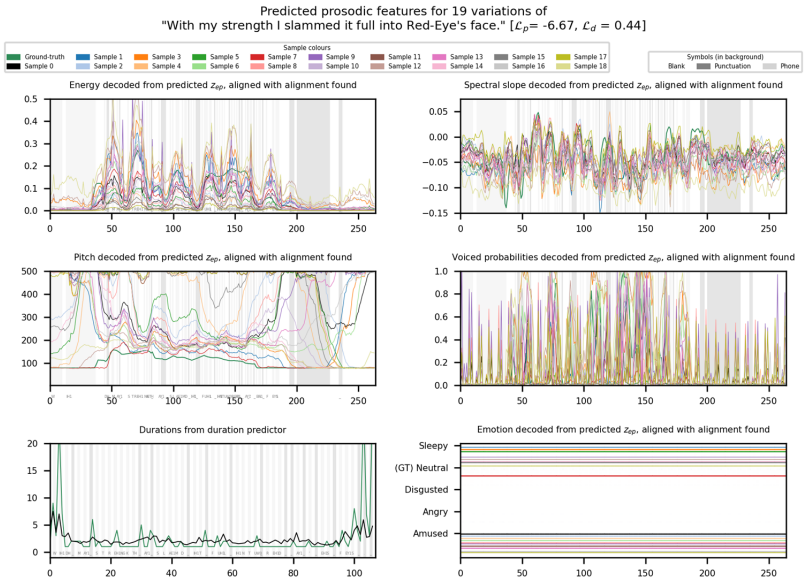
We explain the terms topical emphasis, marked tonicity, and syntactic phrasing as defined in [29].

Topical emphasis can be placed in an utterance by emphasizing a topic through prosodic variation. For example, points for topical emphasis in the utterance *Yesterday I was walking the dog in the park* may be "Yesterday", "I", "dog", and "park".

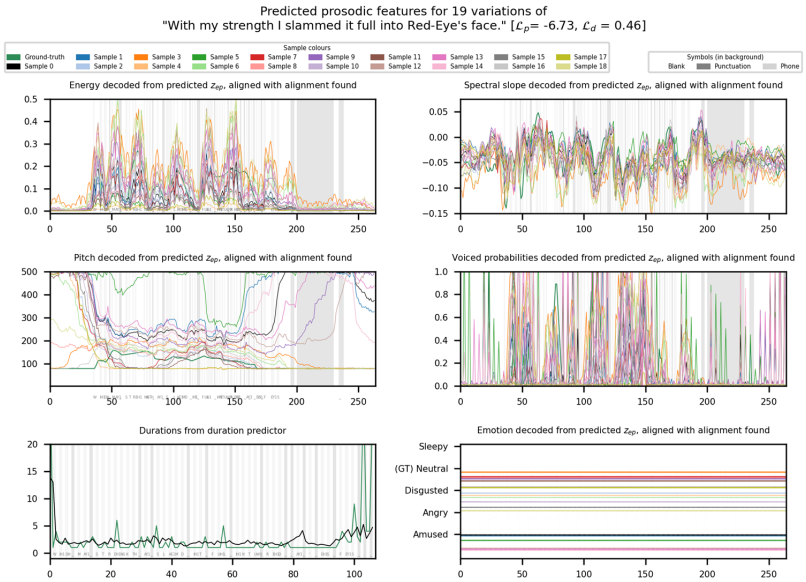
Marked tonicity and syntactic phrasing are both used to disambiguate meaning in an utterance. Syntactically ambiguous sentences can be clarified through tonicity (marked tonicity) or pauses (syntactic phrasing). An example for marked tonicity is *They were milking cows.*. The first interpretation is that "milking" is a verb, making "they" refer to persons. The second interpretation, signaled through specific tonicity on the "milking" word, signals "milking" is an adjective. In that case, "they" refers to cows that are used for milking.

For syntactic phrasing, we use pauses to disambiguate meaning. For example, the utterance *For my dinner I will have either pork or chicken and fries* without extra punctuation carries two possible meanings. Pausing before "and" signals that the meal consists of fries, and next to that pork or chicken. Pausing before "or" signals the choice is between pork without fries and chicken with fries.

B. Prosodic variations, single utterance

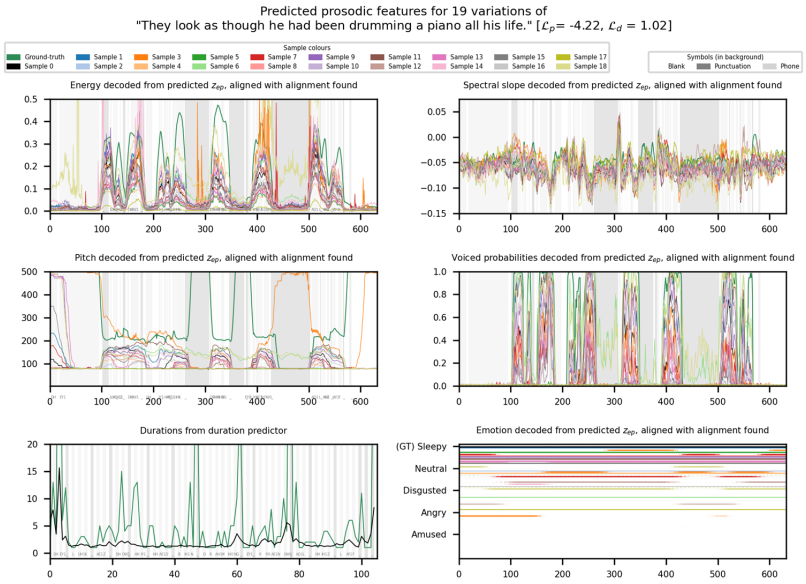


(a) *1.4*

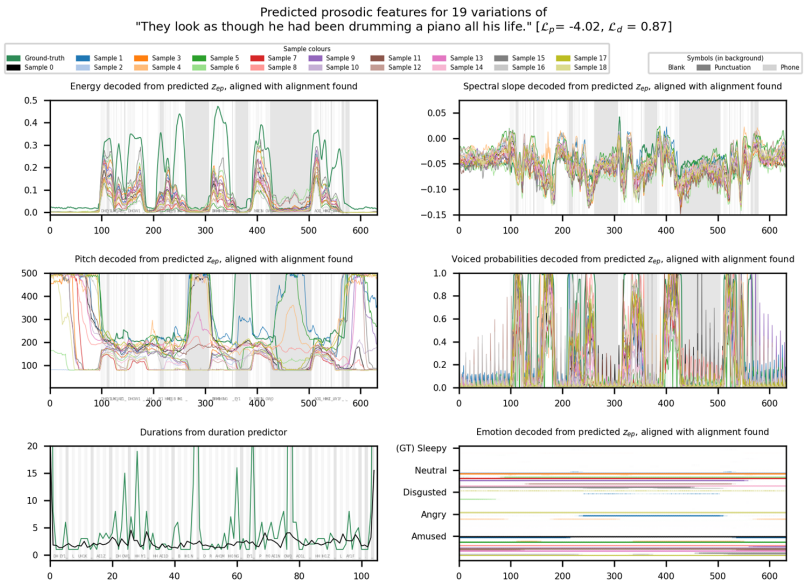


(b) *1.4 switch*

Figure 11: *Prosodic variations for one training utterance. The samples represent extremes of the $\boldsymbol{Z_p}$ dimensions at temperature $T = 2$.*



(c) *1.5*



(d) *1.5 switch*

Figure 11: *Prosodic variations for one training utterance. The samples represent extremes of the $\boldsymbol{Z_p}$ dimensions at temperature $T = 2$. (cont.)*

C. Prosodic latent space clusters

Figure 12: *Prosodic latent space of 1.5 freeze, visualized using PCA. Every point is a timeframe of an EmoV-DB training utterance reverse decoded via the prosody decoder. Colour indicates emotion label.*

D. Quantitative listening test

D.1. MUSHRA interface

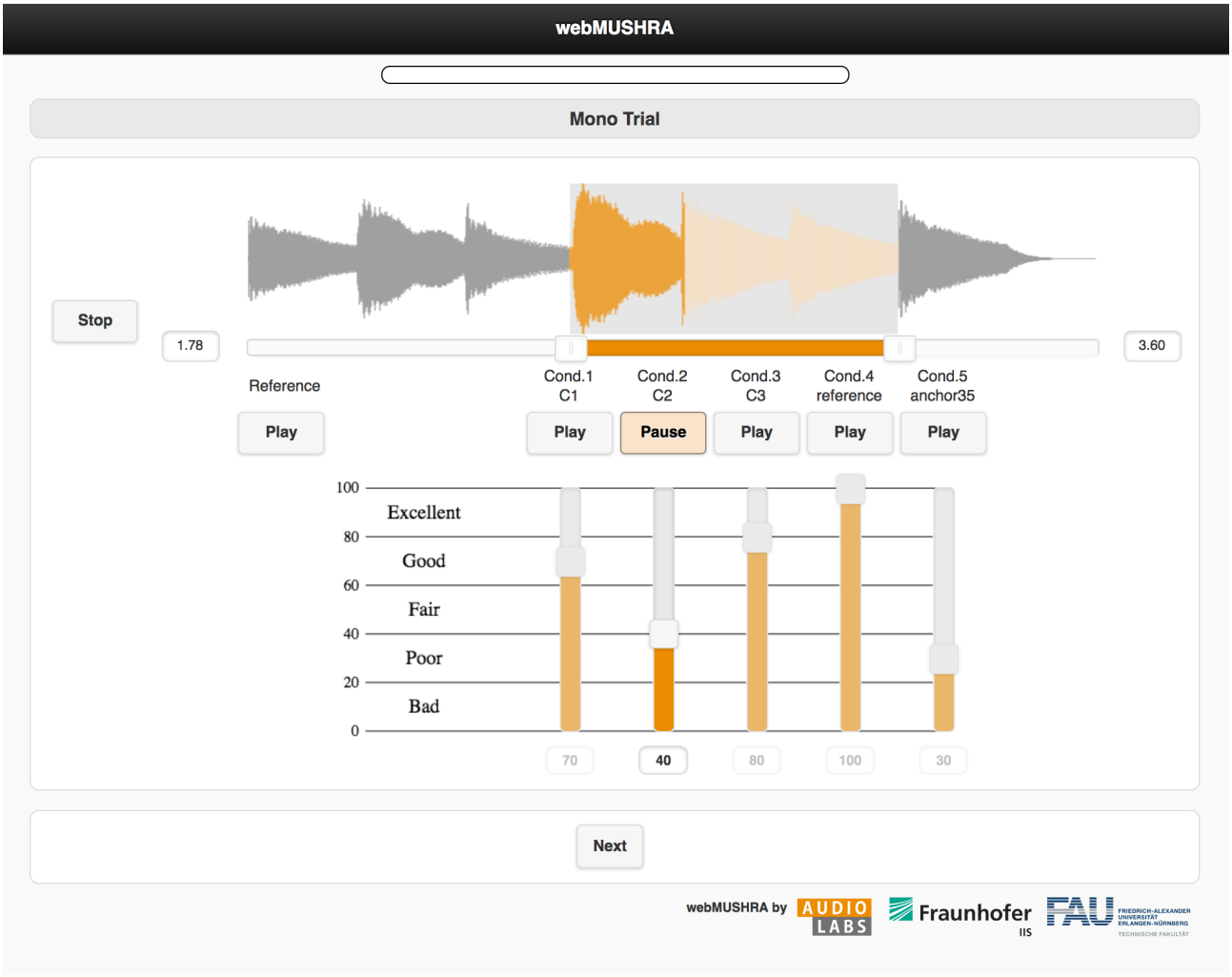


Figure 13: *An example of a webMUSHRA [23] screen. The assessor listens to multiple stimuli and ranks them. It includes a reference sample, a hidden version of the reference, and an anchor.*

D.2. Results

		N	Mean	95% Confidence Lower	95% Confidence Upper	Model vs Base Mean Difference	Sig.
Naturalness*	**Base**	171	45.61	42.26	48.96	**-9.047**	<0.001
	Model	171	36.57	33.39	39.74		
	High anchor	19	75.89	64.40	87.39		
	Reference	38	97.79	95.71	99.87		
Emotion	**Base**	144	16.77	13.02	20.52	5.188	0.247
	Model	144	21.96	18.10	25.81		
	Low anchor	16	55.25	41.83	68.67		
	Reference	32	59.50	49.55	69.45		
Interpersonal attitude	**Base**	171	12.47	9.45	15.48	-0.281	1.000
	Model	171	12.19	9.22	15.15		
	Low anchor	19	32.05	15.57	48.53		
	High anchor	38	32.97	24.53	41.42		
	Reference	38	39.89	29.42	50.37		
Propositional attitude	**Base**	171	12.54	9.52	15.56	-2.211	0.897
	Model	171	10.33	7.65	13.00		
	Low anchor	19	15.11	4.36	25.85		
	High anchor	38	50.03	38.20	61.85		
	Reference	38	56.84	45.39	68.30		
Topical emphasis	**Base**	144	20.72	17.44	24.00	**-7.521**	0.010
	Model	144	13.20	10.34	16.06		
	High anchor	32	79.84	68.67	91.02		
	Reference	32	82.53	74.96	90.10		
Style	**Base**	144	17.74	14.13	21.35	5.347	0.236
	Model	144	23.08	19.25	26.92		
	Low anchor	16	54.25	35.06	73.44		
	Reference	32	83.72	73.14	94.30		
Marked tonicity	**Base**	171	34.70	30.18	39.21	**-9.462**	0.017
	Model	171	25.23	21.67	28.80		
	High anchor	19	50.79	30.69	70.89		
	Reference	38	52.63	38.49	66.77		
Syntactic phrasing	**Base**	144	38.15	34.74	41.56	-3.965	0.368
	Model	144	34.19	30.33	38.04		
	High anchor	16	96.19	89.99	102.39		
	Reference	32	95.84	93.35	98.34		

Table 7: *Expressive diversity results of the user study, i.e. the count of unique subcategories a user annotated across all samples for a system. Firstly, T-tests determined whether sampling at another temperature increases the subcategory diversity – listed in the 'All vs Small' column. A second sequence of T-tests determined whether the model has more unique subcategories than the base. Values in bold are significant at the 0.05 level.*

Category	System	Subset	N	Mean	95% Confidence Lower	95% Confidence Upper	All vs Small Mean Difference	All vs Small Sig.	Model All vs Base All Mean Difference	Model All vs Base All Sig.
Emotion	Base	All	16	3.56	2.95	4.18	0.438	0.644	0.500	0.539
		Small	16	3.13	2.61	3.64				
	Model	All	16	4.06	3.46	4.66	**1.188**	0.012		
		Small	16	2.88	2.36	3.39				
Interpersonal attitude	Base	All	18	3.11	2.48	3.75	0.368	0.438	0.056	0.999
		Small	18	2.56	2.13	2.98				
	Model	All	18	3.17	2.57	3.76	0.778	0.160		
		Small	18	2.39	1.87	2.90				
Propositional attitude	Base	All	18	3.11	2.57	3.65	0.889	0.740	0.444	0.607
		Small	18	2.22	1.82	2.62				
	Model	All	18	3.56	2.87	4.24	0.889	0.740		
		Small	18	2.67	2.18	3.15				
Topical emphasis	Base	All	16	3.06	2.57	3.56	0.438	0.392	-0.187	0.904
		Small	16	2.63	2.20	3.05				
	Model	All	16	2.88	2.55	3.20	0.438	0.392		
		Small	16	2.44	2.05	2.83				
Style	Base	All	16	3.69	3.15	4.23	0.750	0.142	0.063	0.998
		Small	16	2.94	2.58	3.30				
	Model	All	16	3.75	3.09	4.41	0.875	0.064		
		Small	16	2.88	2.40	3.35				
Marked tonicity	Base	All	18	1.89	1.73	2.05	0.167	0.517	0.000	1.000
		Small	18	1.72	1.49	1.95				
	Model	All	18	1.89	1.73	2.05	0.000	1.000		
		Small	18	1.89	1.73	2.05				
Syntactic phrasing	Base	All	16	1.81	1.60	2.03	0.188	0.682	-1.250	0.879
		Small	16	1.63	1.36	1.89				
	Model	All	16	1.69	1.43	1.94	0.125	0.879		
		Small	16	1.56	1.29	1.84				

Table 8: *Intensity scores of the user study. For naturalness, instead of intensity, naturalness is rated. T-tests determine whether model has higher scores than base. Values in bold are significant at the 0.05 level.*

E. Model configuration

Listing 1: *Configuration file of model* `1.4`


```
{
  "train": {
    "use_cuda": true,
    "log_interval": 20,
    "save_interval_epoch": 25,
    "vis_interval": 25,
    "seed": 1234,
    "epochs": 5000,
    "learning_rate": 1e0,
    "betas": [0.9, 0.98],
    "eps": 1e-9,
    "warmup_steps": 4000,
    "start_pretraining": false,
    "scheduler": "noam",
    "batch_size": 32,
    "ddi": true,
    "fp16_run": false,
    "with_tensorboard": true
  },
  "data": {
    "load_mel_from_disk": false,
    "load_ep_from_disk" : true,
    "name": "lemof",
    "text_cleaners":["english_cleaners_minus"],
    "max_wav_value": 32768.0,
    "sampling_rate": 16000,
    "filter_length": 1024,
    "hop_length": 256,
    "win_length": 1024,
    "n_mel_channels": 80,
    "n_p_channels": 4,
    "mel_fmin": 0.0,
    "mel_fmax": 8000.0,
    "pitch_fmin": 80.0,
    "pitch_fmax": 500.0,
    "add_noise": true,
    "add_blank": true,
    "cmudict_path": "g2p/cmu_dictionary_lemof",
    "en-uk": false,
    "emotions": ["Amused", "Angry", "Disgusted", "Neutral", "Sleepy"],
    "emotion_colors": ["orange", "red", "green", "black", "blue"]
  },
  "model": {
    "hidden_channels": 192,
    "filter_channels": 768,
    "filter_channels_dp": 192,
    "gin_channels": 2,
    "ein_channels": 2,
    "kernel_size": 3,
    "p_dropout": 0.1,
    "n_blocks_dec": 12,
    "n_layers_common_enc": 4,
    "n_layers_text_enc": 2,
    "n_layers_prosody_enc": 2,
    "n_layers_merge": 3,
    "n_heads": 2,
    "p_dropout_dec": 0.05,
    "dilation_rate": 1,
    "kernel_size_dec": 5,
    "n_block_layers": 4,
    "n_sqz_spec": 2,
    "n_sqz_ep": 10,
    "latent_p_channels": 9,
    "prenet": true,
    "mean_only": true,
    "hidden_channels_enc": 192,
    "hidden_channels_dec": 192,
    "hidden_channels_merge": 192,
    "window_size": 4,
    "latent_merge": "transform_input",
    "n_speakers": 5,
    "n_emotions": 5
  }
}
```

F. Elaboration of future work

Aggregated prosodic variation plots Our qualitative prosodic variation plots can be enriched with more results. One can aggregate the prosodic variations found from sampling across multiple utterances and for multiple values of sampling temperature T. We can turn to the novel visualization method of Ctrl-P [13]. An example and explanation can be found in Figure 14. Ctrl-P focused on disentangled control pitch, energy, and durations, whereas we focus on exploration. We expect these plots to be more similar to the right subplot of Figure 14.

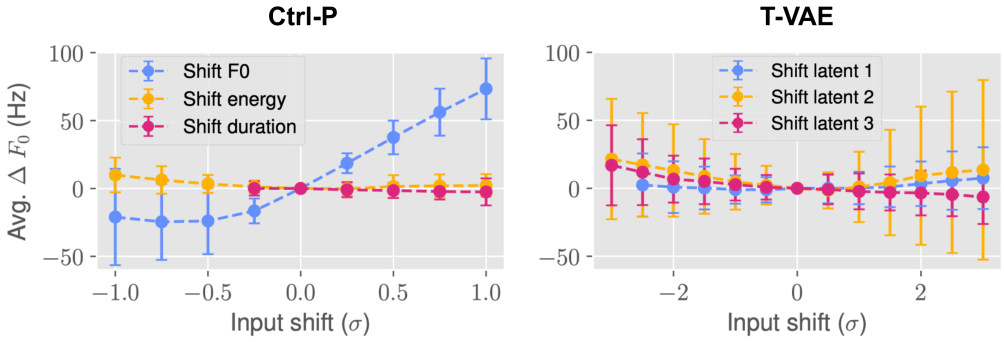


Figure 14: *Example of an aggregated prosodic variation plot, from [13]. It shows the influence of every latent space dimension on one prosodic feature, in this case, pitch. The x-axis shows the value chosen for the dimension d, expressed as the shift from $\boldsymbol{\mu}_d$, measured in σ_d. The pitch, or F0, is analyzed on a global level; it is averaged over timeframes for each utterance. Results are aggregated over all utterances in the validation set. A point indicates the mean across the subset, and the whiskers one standard deviation.*