
Learning De-identified Representations of Prosody from Raw Audio

Jack Weston¹ Raphaël Lenain¹ Udeepa Meepegama¹ Emil Fristed¹

Abstract

We propose a method for learning de-identified prosody representations from raw audio using a contrastive self-supervised signal. Whereas prior work has relied on conditioning models on bottlenecks, we introduce a set of inductive biases that exploit the natural structure of prosody to minimize timbral information and decouple prosody from speaker representations. Despite aggressive downsampling of the input and having no access to linguistic information, our model performs comparably to state-of-the-art speech representations on DAMMP, a new benchmark we introduce for spoken language understanding. We use minimum description length probing to show that our representations have selectively learned the subcomponents of non-timbral prosody, and that the product quantizer naturally disentangles them without using bottlenecks. We derive an information-theoretic definition of speech de-identifiability and use it to demonstrate that our prosody representations are less identifiable than other speech representations.

1. Introduction

To produce and understand spoken language, humans encode and decode audio information at multiple timescales. Phonetic information is used to decode linguistic content, which carries part of the meaning in speech. Another source of meaning, prosody, can be decoded through non-phonetic acoustic patterns; for example to identify who’s speaking, which primarily relies on the timbre subcomponent of prosody (Skerry-Ryan et al., 2018; Qian et al., 2020). Artificial systems mimicking these human processes must solve the similar problems of representing phonetic information (to obtain a linguistic representation) and representing prosodic information (Baevski et al., 2020; Wan et al., 2019).

¹Novoic, London, UK. Correspondence to: Jack Weston <jack@novoic.com>.

The phonetic problem has automatic speech recognition (ASR) as its obvious use-case. In recent years, speech representation learning has been increasingly dominated by self-supervised frameworks using contrastive losses. These include contrastive predictive coding (CPC) (Oord et al., 2018), wav2vec (Schneider et al., 2019), vq-wav2vec (Baevski et al., 2019) and wav2vec 2.0 (Baevski et al., 2020), which learn representations directly from raw audio. These models are generally frame- or phone-based and use fine-timescale, fixed-length audio frames as input. This promotes encoding of high-frequency phonetic information, crucial for transcription, but makes them less incentivized to capture patterns occurring on longer timescales. Other approaches have used triplet loss and temporal proximity as a training signal to learn “semantic” (Jansen et al., 2018) or “non-semantic” (Shor et al., 2020) representations from spectrograms.

The prosodic problem has been less studied. Its primary use-case has been building more expressive text-to-speech (TTS) systems. Prior approaches to learning representations of prosody have relied on subtractive definitions such as (Skerry-Ryan et al., 2018): “Prosody is the variation in speech signals that remains after accounting for variation due to phonetics, speaker identity, and channel effects (i.e. the recording environment)”. These approaches typically use autoencoders conditioned on lexical information and speaker identity (Skerry-Ryan et al., 2018; Wang et al., 2018; Battenberg et al., 2019; Zhang et al., 2019). This encourages the remaining information to be contained in a bottleneck that encodes prosody. Other work has used a triple bottleneck to further decompose prosody in its subcomponents (Qian et al., 2020). For *non-timbral prosody* (i.e. what remains after removing speaker characteristics), these subcomponents are pitch, rhythm and tempo, acoustically reflected in the fundamental frequency (F_0 , the pitch contour), intensity or energy (c_0), and the speech rate respectively. WaveNet makes explicit use of F_0 , c_0 and phone durations to synthesize speech (Oord et al., 2016). CHIVE represents prosody using F_0 , c_0 and phoneme durations as features in a conditional variational autoencoder (Wan et al., 2019). These models share a set of characteristics that motivate our work: they are subtractive and rely on conditioning models on bottlenecked information. Having been developed in the context of TTS, they have inductive

biases that are particularly suited to learning primarily phonetic representations, rather than prosodic representations, a weakness highlighted by Oord et al. (2016).

The contributions of this paper are as follows:

- Introducing and characterizing VQP, a self-supervised contrastive model that learns to selectively represent non-timbral prosody from raw audio without using bottlenecks.
- Adapting probes from the natural language processing (NLP) literature to demonstrate that VQP representations selectively encode the subcomponents of non-timbral prosody.
- Demonstrating that product quantization can be used for disentanglement of audio representations without using bottlenecks.
- Introducing an information-theoretic definition of de-identification using prequential probes on a speaker verification task.
- Benchmarking a number of state-of-the-art, self-supervised audio representation learning models on a set of tasks for spoken language understanding, as well as quantifying their de-identifiability.

2. Approach

2.1. Data Preprocessing

Before data is passed to the model, it undergoes a series of preprocessing steps. The raw audio is first resampled to 16 kHz, then pitch-shifted on a per-example basis such that the median pitch of the voice segments is the same value for the whole dataset. To perform this pitch-shifting, we ran an autocorrelation-based pitch-tracking method (via Praat (Boersma, 2006)), calculated the median pitch of the voiced segments in a given sample, then shifted the pitch such that the median is 150 Hz for the sample. This is primarily to address the distributional difference between sexes in fundamental frequency. We find that this makes the resulting representations less identifiable and improves training robustness. The waveform is aggressively downsampled to 500 Hz, retaining the frequency range of the typical fundamental frequency for human speech, but discarding the formants that characterise the speaker’s voice. The highest typical F_0 for humans is ~ 250 Hz (Takefuta et al., 1972) and by applying Nyquist’s theorem (Nyquist, 1928) we choose a sampling rate of 500 Hz. The waveform is normalized to zero mean and unit variance, then sliced into variable-length audio-words using word-level timestamps obtained via ASR or forced alignment. This is based on the intuition that meaningful prosody states are naturally

discretized on a per-word basis. By contrast, wav2vec, vq-wav2vec, wav2vec 2.0, TRILL, Tacotron and CHiVE all use fixed-length audio input (Schneider et al., 2019; Baevski et al., 2019; 2020; Shor et al., 2020; Wang et al., 2017; Wan et al., 2019). In Stehwien & Vu (2017), word-level timestamps are used with spectrogram inputs to a CNN on a classification task to recognise prosodic events. Prior models, developed for ASR or TTS use shorter timescale audio inputs (Baevski et al., 2019; Skerry-Ryan et al., 2018). We include up to 2 seconds of leading silence before each audio word since pause information is important for parts of prosody such as the rhythm and tempo of speech.

2.2. Model

Our model comprises two parts: a prosody encoder and a Transformer encoder (see Figure 1). The prosody encoder $f : \mathcal{A} \mapsto \mathcal{P}$ maps variable-length raw audio $A_{0,t}$, corresponding to a single audio-word, to a fixed-length quantized vector P_t . The sequence of latent prosody representations P_t is fed to a Transformer $f : \mathcal{P} \mapsto \mathcal{C}$ to produce contextualized prosody representations C_t that can capture information from the entire sequence, unlike the audio-word-level P_t representations. Our core hypothesis is that prosody has predictable temporal patterns, occurring at frequencies lower than 250 Hz that can be learned directly from the acoustic signal. We use a contrastive, self-supervised signal, similar to Baevski et al. (2020), where only raw audio is used as the input and target. Unlike subtractive approaches to representing prosody (Skerry-Ryan et al., 2018; Wan et al., 2019), our model does not rely on lexical inputs. Instead, it only has access to the downsampled raw audio signal and word-level timestamps.

Temporal convolutional network: The first module of the prosody encoder is a temporal convolutional network (TCN) (Oord et al., 2016; Bai et al., 2018), comprising a stack of causal dilated 1D convolutions with residual connections, which we adapt with skip connections (He et al., 2016). The strides, number of layers and kernel sizes are chosen such that the receptive field of the TCN spans the maximum sequence length of one audio word. Inspired by the WaveNet architecture (Oord et al., 2016), we use summed skip-connections as the TCN output rather than the output of the final layer to allow the network to more easily capture features with different time-resolutions. To reduce across the temporal (frame) dimension, we max-pool the skip matrix, which we empirically found led to more robust training than mean pooling or selecting the final non-padded element in the skip matrix for each element in the batch. We exploit the exponentially increasing receptive field of our TCN to capture the longer-range dependencies that encode prosodic information. By contrast, other approaches to encoding prosody have relied on conditional autoencoders (Skerry-Ryan et al., 2018; Wan et al., 2019; Zhang et al.,

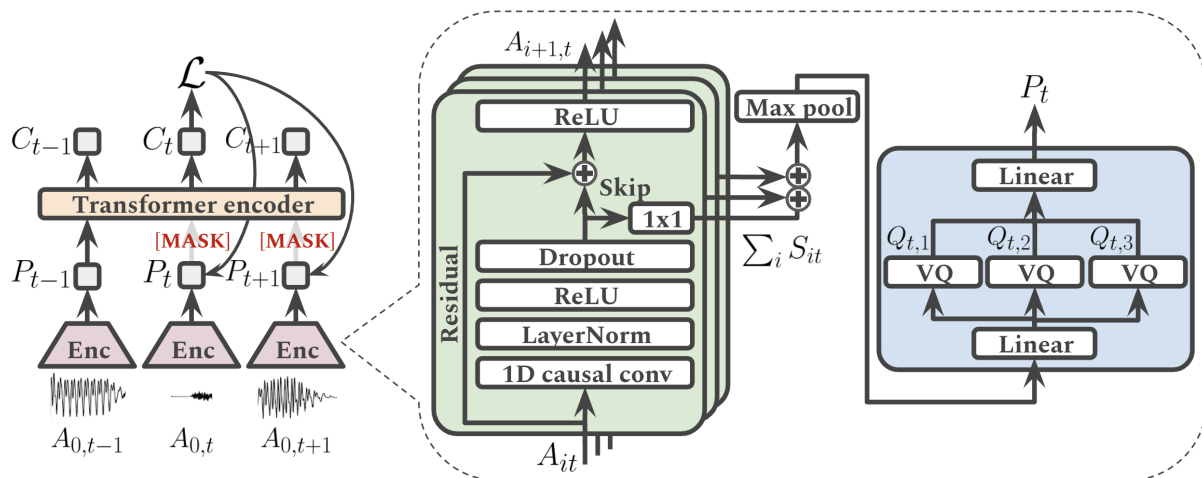


Figure 1. The model architecture, including an overview of the entire model (left) and a focus on the prosody encoder (right).

2019) or CNNs (Stehwien & Vu, 2017). TTS-motivated models encoding phonetic information have used linguistic features such as F_0 , duration and c_0 (Wan et al., 2019; Oord et al., 2016). ASR-motivated models have used raw audio passed into a TCN (Baevski et al., 2020) or a vanilla CNN (Baevski et al., 2019).

Product quantizer: The max-pooled output of the TCN is passed to a product quantizer, whose constituent vector quantizers are inspired by VQ-VAE-2 Razavi et al. (2019) but adapted for product quantization. The product quantizer itself is similar to wav2vec 2.0, wherein the input data undergoes an affine transformation before having its features sliced into M equal parts, all of which are passed to a vector quantizer. Following quantization, the output vectors are concatenated and undergo a final affine transformation. Following (Razavi et al., 2019), each constituent vector quantizer learns a nonlinear mapping from its input space S to a vector $E(s)$, which is replaced with the nearest prototype vector in the codebook $e_k, k \in 1, \dots, K$:

$$\text{Quantize}(E(s)) = e_k, \quad (1)$$

where $k = \arg \max_j \|E(s) - e_j\|$. This mapping is learnt via backpropagation using the straight-through gradient estimator (Bengio et al., 2013). Using multiple vector quantizers is not equivalent to using one with a larger capacity; the inclusion of affine transformations before and after the vector quantization gives the network some capacity to map the input data into a more convenient basis before slicing. We will explore whether product quantization can be used to disentangle the space of representations and add explainability to representation learning models. The use of VQ-VAE-style quantization over the Gumbel softmax (Jang et al., 2016) approach used in wav2vec 2.0 was an empirical deci-

sion to improve training robustness. This is consistent with the claim in the original VQ-VAE paper (Oord et al., 2017) that their method experiences less gradient variance during training. We deliberately restrict the number of quantized states in our codebook while learning the vector-quantized representations to encourage representations to be parsimonious and avoid “hiding” nuisance covariates, which may include speaker-identifiable information, in small details.

Transformer encoder: The product-quantized vector sequence is fed to a standard Transformer encoder architecture (Vaswani et al., 2017). We use fixed sine/cosine positional embeddings to allow the encoder to exploit position information. In wav2vec (Schneider et al., 2019), representations of audio data are learned by solving a self-supervised context-prediction task with the same loss function as word2vec (Mikolov et al., 2013). One aim of contextualizing prosody representations is to make representations with weaker cross-temporal interactions, which may facilitate audio-linguistic representation learning in future work. Context-aware representations of time-series often make better predictions (Peters et al., 2018; Devlin et al., 2018; Oord et al., 2018) and we hypothesise that contextualization makes stronger prosodic representations.

2.3. Probing and De-identification

For explainability, we aim to measure how well a feature is represented in a given representation. We use the prequential (or online) approach to minimum description length (MDL) probing to quantify the regularity between representations and labels (Rissanen, 1978; Voita & Titov, 2020). Formally, MDL measures the number of bits required to transmit the labels given the representations. If a feature

is easily extractable from a given representation, a model trained to detect said feature will converge quickly, resulting in a small MDL. Computing the MDL using the prequential approach requires sequential training and evaluation. We partition the train set, $D = \{(x_i, y_i)\}_{i=1}^n$, into timesteps, $1 = t_0 < t_1 < \dots < t_S = n$, and train our probe, $p_\theta(y|x)$, such that at timestep t_i the train set is $\{(x_j, y_j)\}_{j=1}^{t_i}$ and we evaluate on set $\{x_j, y_j\}_{j=t_i+1}^{t_{i+1}}$, calculating the codelength as per Voita & Titov (2020).

We further adapt this method to derive an information-theoretic definition of speech identifiability. Following the literature (Tomashenko et al., 2020), we consider a number of binary speaker verification trials but, instead of using equal error rate or log-likelihood-based metrics, we define the *de-identification ratio* of a set of trial representations $\{s_i\}$ with respect to enrolment representations $\{r_i\}$ as the inverse of the compression ratio of the theoretical minimum description length to transmit the data using a prequential approach:

$$\mathcal{D}(s, r, M(\theta), D) = \frac{t_1}{n} - \frac{1}{n} \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | r_{t_i+1:t_{i+1}}, s_{t_i+1:t_{i+1}}). \quad (2)$$

A full derivation is given in the supplementary materials. The rationale is that a shorter MDL means that the verification task is easier given the two representations. This improves upon prior work, which assumes a fixed model (usually a probabilistic LDA (Tomashenko et al., 2020; Han et al., 2020)), by taking into account the effort required to perform verification as well as performance on the task. Real attackers could have access to sophisticated models and arbitrary computational resources to compare speech representations, motivating this approach. Prior work performs verification on pairs of i-vectors (Tomashenko et al., 2020); we likewise consider pairs of the same representation, but note that cross-representation comparisons ought to be included in more comprehensive studies, including raw audio and spectrogram as inputs. For simplicity, we mean-pool sequential representations over time but note that this could underestimate the identifiability of the sequence as a whole due to lost information.

3. Experimental Setup

3.1. The Colossal Audio-Linguistic Corpus (CALC)

Models using self-supervised pretraining consistently demonstrate the importance of large datasets (Devlin et al., 2018; Raffel et al., 2019; Brown et al., 2020). AudioSet (Gemmeke et al., 2017), for instance, is a large dataset for general-purpose audio machine learning, a significant subset of which has speech tags. For pretraining our mod-

els, we construct the Colossal Audio-Linguistic Corpus (CALC), a large word-aligned audio-linguistic dataset of natural speech with matching audio and text modalities. CALC is composed of five datasets wrangled into a common format, chosen based on their size, prior use in the literature, and whether they contain natural speech as opposed to read speech (Table 1). Except for the AMI dataset, which already has word-level alignment (Carletta et al., 2005), we performed alignment using the Montreal Forced aligner (McAuliffe et al., 2017). The full CALC dataset contains ~ 7.5 million words of natural speech.

3.2. The DAMMP Benchmark for Spoken Language Understanding

To standardize assessment of representations for spoken language understanding, we introduce a new benchmark, DAMMP. The dataset has parallel audio and text modalities of natural speech, so audio-based, text-based and audio-linguistic models can be benchmarked. DAMMP is composed of five datasets (Table 1) all with binary classification tasks where prosody is important: DAIC-WOZ (Low et al., 2020), ADReSS (de la Fuente Garcia et al., 2020; Pompili et al., 2020), MUStARD (Bryant, 2010; Woodland & Voyer, 2011), CMU-MOSEI (Liu et al., 2018; Jain et al., 2018), and POM (Okada et al., 2016; Siddiquie et al., 2015). As with CALC, we performed word-level alignment for all datasets using the Montreal Forced aligner (McAuliffe et al., 2017). DAIC-WOZ, ADReSS, and CMU-MOSEI already had a canonical test set disjoint by speaker, whereas for MUStARD and POM we sampled the datasets to make balanced test sets for the binary variable of interest. For MUStARD, the train/test sets are disjoint by TV show as well as speaker, to make the task harder. DAIC-WOZ, ADReSS and MUStARD already had canonical binary labels to predict. For CMU-MOSEI and POM, we converted Likert-scale ratings (persuasiveness and sentiment respectively) to binary labels, following Park et al. (2014).

3.3. Baselines

We compare VQP’s performance on DAMMP and on de-identification with four recent audio representation learning models: TRILL (Shor et al., 2020), wav2vec 2.0 (Baevski et al., 2020), vq-wav2vec (Baevski et al., 2019) and Mockingjay (Liu et al., 2020). We chose these baselines as they each bear similarity to VQP in different ways, though we note that we were unable to obtain a pure prosodic baseline. An improvement for future work would be to compare with a baseline from a TTS model, such as Skerry-Ryan et al. (2018); Wan et al. (2019); Wang et al. (2018). Additionally, we compare against x-vectors (Snyder et al., 2018; Ravanelli et al., 2021) which are specifically trained to learn identifiable representations through speaker identification pretraining.

Learning De-identified Representations of Prosody from Raw Audio

CALC	DAMMP	Dataset	Target	Description	Size	Ref.
-	✓	DAIC-WOZ	Depression diagnoses	Interviews by a virtual interviewer	~300 interviews	(Gratch et al., 2014)
-	✓	ADReSS	Alzheimer’s disease diagnoses	Picture description tasks	~200 descriptions	(Luz et al., 2020)
-	✓	MUStARD	Sarcasm labels	Acted scenes from TV shows	~6.4k utterances	(Castro et al., 2019)
✓	✓	CMU-MOSEI	Sentiment labels	Spoken product reviews	~20k utterances from ~2k speakers	(Zadeh et al., 2018)
✓	✓	POM	Persuasiveness labels	Film reviews	~300 reviews	(Park et al., 2014)
✓	-	TED-LIUM 3	-	TED talks	~2.4k TED talks	(Hernandez et al., 2018)
✓	-	LRS2	-	Single utterances from BBC TV scenes	~140k utterances	(Afouras et al., 2018)
✓	-	AMI	-	Real and acted meetings	~100 hours of meetings	(Carletta et al., 2005)

Table 1. The source datasets for the Colossal Audio-Linguistic Corpus (CALC) and the DAMMP benchmark for spoken language understanding, along with a short description of the nature and size of the data. CMU-MOSEI and POM feature in both datasets due to their size and the existence of relevant targets. The DAMMP benchmark defines a set of supervised classification tasks, the targets of which are shown in the table.

3.4. Training

The model is trained using a self-supervised contrastive signal, followed by assessing performance on a supervised task. The representations are not fine-tuned on the supervised task to preclude the model from pulling out new, perhaps identifiable, information from the raw audio during supervision. With that caveat, we will still refer to the self-supervised step as ‘pretraining’ for convenience.

We pretrain using a BERT-like masking paradigm, with a contrastive self-supervised signal similar to wav2vec 2.0. The pretraining task is to identify the correct latent prosody representation in the presence of a number of distractors sampled from other masked timesteps. We mask timesteps with a fixed probability and consider a two-part loss function: a contrastive loss and a commitment loss,

$$\mathcal{L} = \mathcal{L}_{\text{contrast}} + \alpha \mathcal{L}_{\text{commit}}, \tag{3}$$

where α is a tunable constant. The contrastive loss for selecting the true latent prosody representation \mathbf{q}_t amongst a set of K distractors $\tilde{\mathbf{q}}_t \in \tilde{\mathbf{Q}}_t$, which are uniformly sampled from other masked timesteps of the same sample, is given by:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp[\mathbf{c}_t^T \mathbf{q}_t / (\kappa \|\mathbf{c}_t\| \|\mathbf{q}_t\|)]}{\sum_{\tilde{\mathbf{q}}_t \in \tilde{\mathbf{Q}}_t} \exp[\mathbf{c}_t^T \tilde{\mathbf{q}}_t / (\kappa \|\mathbf{c}_t\| \|\tilde{\mathbf{q}}_t\|)]}, \tag{4}$$

where κ is a tunable constant. The commitment loss penal-

izes discrepancies between the quantizer inputs and outputs to encourage robustness. We average the commitment loss over the N constituent vector quantizers in our product quantizer:

$$\mathcal{L}_{\text{commit}} = \frac{1}{N} \sum_{i=1}^N \|\text{sg}[\mathbf{e}_i] - E_i(\mathbf{x})\|_2^2, \tag{5}$$

where $\text{sg}(\cdot)$ is the stop gradient operator and \mathbf{x} is the training example. In lieu of a codebook loss, we use exponential moving average updates for the codebook as per Oord et al. (2017).

For training on downstream tasks, we use a simple two layer feed-forward network (FFN) with hidden size 256, batch size 256, ReLU activations, dropout with probability 30% using the Adam optimizer (Kingma & Ba, 2014) with learning rate $\alpha = 10^{-3}$ and default parameters $\beta_1 = 0.9, \beta_2 = 0.99$. We use a final sigmoid nonlinearity and binary cross-entropy loss. The input dimension varies across the different representations. We train these models for 20k steps and use the last model states to report performance on the downstream tasks.

3.5. Experiments

We pretrain our models using a proprietary framework built on top of PyTorch (Paszke et al., 2019). We uniformly

mask 30% of all prosody tokens. The TCN comprises 9 layers, each with 30 filters, a stride of 1 and a kernel size of 2. We use exponentially increasing dilations of size 1, 2, 4, 8, 16, 32, 64, 128, 256 to yield a receptive field size of 512 frames. The 1×1 convolution similarly has 30 filters. The dropout probability is 10%.

The product quantizer comprises 3 vector quantizers each of dimension 10 with an independent codebook of size 32, giving a maximum number of states of $32 \times 32 \times 32 \approx 32.8\text{k}$ per audio-word. We experiment with fewer and more states (see supplementary materials). We choose a decay of $\gamma = 0.99$ for all quantizers and weight the commitment loss by $\alpha = 0.5$. The linear layers have dimensionality 30.

The Transformer encoder has 12 layers, 12 attention heads, inner (FFN) dimension 3,072, embedding size 768, ReLU activation and a 10% dropout probability. The positional encoding is implemented as per Vaswani et al. (2017). We postulate that prosody temporal interactions are relatively short compared to language and restrict the sequence length to 32 words. During pretraining, we also require a minimum sequence length of 16 words. We train using $K = 9$ distractors.

We linearly warm up the learning rate from 0 to a maximum of 1.5×10^{-5} at 10k steps before linearly decaying it to 0 at the step. The model trains for 250k steps using the AdamW optimizer (Loshchilov & Hutter, 2017). We use a batch size of 128 samples and train on a single V100 GPU for 2.3 days.

4. Results and Discussion

4.1. Benchmarking Performance and De-identifiability

The results on DAMMP and the de-identifiability task are given in Table 2 for this work (VQP), baselines and ablations. No model uniformly performed the best across all benchmark tasks. VQP performed the best on DAIC-WOZ (AUC = 0.667), TRILL the best on ADReSS (AUC = 0.770), wav2vec-2.0 on MOSEI (AUC = 0.835), vq-wav2vec on MUSStARD (AUC = 0.618), and x-vector on POM (AUC = 0.877). Averaging the AUCs across all datasets (which we term the DAMMP score), vq-wav2vec performed the best (score = 0.689) and TRILL comparably (score = 0.687). VQP had an average score of 0.633. The observation that different representations performed the best on different tasks highlights the complexity of spoken language understanding, and the scope for improved representations in the area.

VQP had the worst performance compared to other models on MOSEI and MUSStARD. For MOSEI, examples are single utterances of 1 – 20 words, about an order of magnitude shorter by words than the other tasks. Since VQP produces

word-level representations whereas other baselines work at a finer timescale, we hypothesize that the time-averaging did not have enough examples to produce a high-quality pooled representation for VQP. For MUSStARD, we hypothesize that the presence of laughter following sarcastic remarks might be used as a training signal for the baselines (which consider all audio), whereas VQP only has access to audio that corresponds to spoken words and up to 2 seconds of preceding silence, which neglects the laughter which primarily occurs after the final word.

Our ablations show that contextualization improved VQP performance, providing support for our belief that prosody is context-dependent. Pitch scaling had a significant effect on performance, potentially through making the model converge earlier. Removing the product quantizer only led to a small drop in average AUC but, dissecting further, performance dropped substantially on all tasks, except for MUSStARD and MOSEI, where it improved. Only quantizing the targets significantly hurt performance, which is notable since it had a positive impact in the original wav2vec-2.0 model (Baevski et al., 2020). Some of these effects could be reflections of causing the models to converge faster or slower and training models for a fixed number of updates.

VQP outperforms the baselines as the most de-identified, with a codelength of 51.06 kbits, de-identification ratio 1.10 and a probe AUC of 0.73, compared to the most identifiable baseline, x-vector, that has a codelength of 16.16 kbits, a DIR of 0.36, and an AUC of 0.99. With the caveat that using performance metrics like AUC directly makes conclusions more strongly model-dependent, we can create a simulation by assuming that we had a speech representation from each of a group of N people and wish to find out from whom a separate target speech representation came. For simplicity, we assume that the model outputs a binary value, the trials are independent and that we must uniquely identify the correct person. Using the classifiers’ positive and negative predictive values, we have that $P_{id}(N) = PPV \times NPV^{N-1}$. For $N = 10$ people, VQP would have a probability of correctly identifying the speaker of 1.58%, TRILL 5.10%, vq-wav2vec 24.8%, wav2vec-2.0 37.7%, Mockingjay 44.3% and x-vector 66.1%.

Our ablations demonstrate that contextualization helps with de-identification, perhaps by building more abstract representations of prosody that retain less of the identifiable low-level information, such as absolute pitches and tempos. Using cross-sample negatives dramatically hurts de-identifiability, in line with our expectations: if trained on negative examples from other speakers, the representations are encouraged to learn more speaker-specific information. Pitch scaling only had a small effect on de-identifiability, contrary to our expectation. Both “no quantization” and “quantizing target only” appeared to improve

Learning De-identified Representations of Prosody from Raw Audio

	DAIC- WOZ	MOSEI	MUS _t ARD	POM	ADReSS	DAMMP score	DIR
VQP (ours)	0.667	0.659	0.488	0.700	0.652	0.633	1.10
TRILL	0.619	0.819	0.546	0.684	0.770	0.687	0.90
wav2vec-2.0	0.541	0.835	0.586	0.722	0.621	0.661	0.56
vq-wav2vec	0.521	0.831	0.618	0.792	0.685	0.687	0.84
Mockingjay	0.463	0.829	0.504	0.785	0.580	0.632	0.60
x-vector	0.491	0.778	0.567	0.877	0.540	0.651	0.36
<u>Ablations</u>							
VQP-PE	0.537	0.643	0.489	0.647	0.64	0.591	0.96
VQP-CSN	0.491	0.694	0.568	0.51	0.295	0.512	0.56
VQP-NPS	0.496	0.63	0.551	0.702	0.639	0.604	1.05
VQP-NQ	0.587	0.69	0.564	0.668	0.641	0.630	1.15
VQP-TQ	0.580	0.510	0.523	0.681	0.607	0.580	1.18
<u>500 Hz filter experiments</u>							
TRILL	0.405	0.526	0.522	0.375	0.500	0.466	1.04
wav2vec-2.0	0.452	0.554	0.547	0.495	0.623	0.534	1.00
vq-wav2vec	0.693	0.556	0.499	0.444	0.478	0.534	1.03
Mockingjay	0.498	0.561	0.512	0.486	0.518	0.515	1.01

Table 2. The results of our work (VQP), baseline representations and ablations on the DAMMP benchmark and the de-identification ratio (DIR). We report AUCs for each constituent classification task and define the DAMMP score as the average. VQP-PE = VQP using the prosody encoder output rather than the Transformer output; VQP-CSN = VQP where the negatives prosodies are cross-sampled, i.e. from other utterances (usually other people); VQP-NPS = VQP trained on data without the median pitch scaling preprocessing step; VQP-NQ = VQP with no product/vector quantization; VQP-TQ = VQP with only targets quantized (similar to wav2vec-2.0).

de-identification. One explanation of this is that the models generally predict less well, which is substantiated by the ablation results on DAMMP, highlighting the trade-off between performance and identifiability. Contrary to our expectations, quantization appears to damage de-identification performance. We hypothesize that this is due to the non-quantized networks having more complex training dynamics, as we observed quantization slowed convergence dramatically.

One ablation that couldn't be done without materially altering the architecture of VQP due to GPU memory constraints, was the importance of the input downsampling. Instead, we applied the 500 Hz filter to the DAMMP data before upsampling back to 16 kHz, which we then used for the downstream tasks of the baselines (Table 2). We observed near-random performance on the DAMMP benchmark and high de-identifiability but we believe this was because the preprocessing incurred too great a domain shift for the baselines, evidenced by the destruction of non-timbral prosodic as well as timbral information, which we probed via an explainability study similar to Section 4.2.

In Figure 2, we investigate the tradeoff between performance and de-identifiability. While wav2vec-2.0 and x-vector perform strongly on the DAMMP benchmark, they

are significantly more identifiable than VQP, TRILL and vq-wav2vec. Similarly, while VQP and its variants tend to be much more de-identified, this is traded off with a suppressed performance. The optimal representation for a given context depends jointly on the complexity of the task and the level of privacy required.

4.2. Using MDL Probes for Explainability

Table 3 summarises the AUCs and minimum description lengths (MDLs) of probes, trained to predict subcomponents of prosody from different representations in the form of audio features (feature distributions are given in the supplementary materials). Whereas the baselines primarily encode timbral information, VQP selectively learns non-timbral prosody. This provides support for the hypotheses that underlie our inductive biases. Both primarily timbral and non-timbral prosodic representations obtain comparable performance on DAMMP, highlighting that there are multiple viable strategies for solving spoken language understanding tasks. VQP encodes complementary, new information compared with other speech representations.

We finally compare the representations in each part of the product quantizer, which is shown in Table 4. Probes trained on contextualized VQP representations showed a similar pat-

Learning De-identified Representations of Prosody from Raw Audio

	TRILL		wav2vec-2.0		vq-wav2vec		Mockingjay		VQP	
	AUC	MDL	AUC	MDL	AUC	MDL	AUC	MDL	AUC	MDL
Pitch										
Pitch	0.558	63.65	0.546	63.88	0.569	63.49	0.558	63.62	0.742	55.78
Rhythm										
Intensity	0.596	63.48	0.557	64.19	0.567	64.10	0.558	64.20	0.662	60.97
Num. sylls	0.519	65.51	0.508	65.58	0.516	65.48	0.513	65.50	0.616	63.13
Tempo										
Artic. rate	0.522	65.19	0.506	65.26	0.514	65.19	0.510	65.29	0.537	65.12
Speech rate	0.532	64.94	0.515	65.03	0.519	64.97	0.519	65.01	0.541	64.88
Syll duration	0.524	65.44	0.509	65.52	0.513	65.48	0.508	65.49	0.497	65.47
Word duration	0.544	65.40	0.522	65.58	0.539	65.47	0.536	65.50	0.749	54.58
Timbre										
Formant f1	0.735	58.03	0.668	62.73	0.696	61.26	0.629	64.07	0.574	65.58
Formant f2	0.743	57.43	0.643	63.11	0.666	62.95	0.586	64.87	0.514	65.60
Formant f3	0.779	54.39	0.667	62.24	0.688	61.92	0.623	63.90	0.509	65.71

Table 3. The explainability results from probing the VQP representations and the baselines. We probe by performing higher-than-mean/lower-than-mean binary classification on a set of speech features. We report the minimum description length (MDL) for each classification, along with the AUC on the final tranche of data using the prequential approach for comparison. For AUC, a higher value suggests the information is better-represented. For MDL, lower values suggest this. We indicate in bold the minimum MDL values for each speech feature.

	VQP-VQ1		VQP-VQ2		VQP-VQ3		VQP-PE		VQP	
	AUC	MDL	AUC	MDL	AUC	MDL	AUC	MDL	AUC	MDL
Pitch										
Pitch	0.588	63.01	0.801	50.04	0.586	63.07	0.870	42.13	0.742	55.78
Rhythm										
Intensity	0.682	59.15	0.681	59.58	0.640	61.49	0.797	50.01	0.662	60.97
Num. sylls	0.694	60.23	0.591	64.29	0.587	64.15	0.805	51.29	0.616	63.13
Tempo										
Artic. rate	0.551	65.00	0.522	65.25	0.536	65.17	0.664	62.19	0.537	65.12
Speech rate	0.555	64.71	0.553	64.48	0.560	64.57	0.696	60.20	0.541	64.88
Syll duration	0.541	65.29	0.530	65.49	0.529	65.53	0.678	61.32	0.497	65.47
Word duration	0.805	51.36	0.735	57.35	0.690	59.41	0.988	14.55	0.749	54.58
Timbre										
Formant f1	0.537	65.86	0.555	65.65	0.537	65.96	0.609	64.24	0.574	65.58
Formant f2	0.516	65.64	0.523	65.62	0.516	65.65	0.547	65.28	0.514	65.60
Formant f3	0.513	65.71	0.510	65.70	0.512	65.66	0.529	65.44	0.509	65.71

Table 4. The disentanglement results from probing the VQP representations. We probe each of the three vector quantizer outputs (VQP-VQ1, VQP-VQ2 and VQP-VQ3) independently as per the setup in Table 3, and recapitulate the VQP results for ease of comparison. We also show the probing results on the full output from the prosody encoder (VQP-PE).

tern to the non-contextualized representations but generally have larger MDLs. We hypothesize this is because the “raw” prosodic information becomes more abstracted after the Transformer. The representations differed in how well the different prosodic features could be predicted from them, and importantly the features clustered based on prosodic

domain for the different representations. Without using bottlenecks, the subcomponents of prosody naturally disentangle. Rhythm and tempo features were best predicted from VQP-VQ1, while pitch was best predicted from representation VQP-VQ2, corresponding to a dissociation of time and frequency features. Median word intensity was

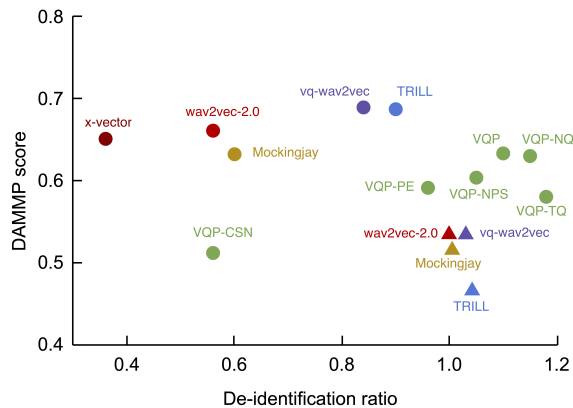


Figure 2. Comparison of VQP, baseline and ablation representations on the DAMMP benchmark and de-identifiability ratio. Triangles represent results obtained using our baselines after downsampling the audio to 500 Hz before upsampling back to 16 kHz. See Table 2 for a definition of the VQP ablation abbreviations and the DAMMP score.

predicted similarly across the representations. Except for word duration, tempo features were not well represented by either factor representation, but notably they were still well represented by the combined representation from the product quantizer, demonstrating that encoding of prosody happens at different levels in the model, especially for these ‘derivative’ features that require a combination of syllable and duration information. VQP-VQ3 was not immediately interpretable. It might be capturing something that is not represented in the features or act in a supporting function for the other representations. Despite downsampling the input, timbral features could be predicted to some degree, which could be due to an association with pitch not removed by the pitch correction or the 500 Hz limit in downsampling still capturing some spectral information; this is supported by the progressively worsening of performance with increasing formant number and that timbral features are predicted by VQP-VQ2, which also represents pitch the best.

5. Conclusions

In this work, we introduced a self-supervised contrastive model that learns to selectively represent non-timbral prosody from raw audio without using bottlenecks, differing from prior subtractive approaches. Our approach retains competitive performance while making the prosody representations more de-identified than prior audio representations, which we quantify using an information-theoretic approach. We make explainability-related contributions by adopting probes for audio representations and demonstrating that product quantization can be used for disentanglement of

prosody without using bottlenecks. Our work is motivated by building better real-world systems for spoken language understanding, where sufficient de-identifiability is crucial for user privacy and complying with HIPAA/GDPR regulation. One natural extension of our work is its application to building audio-linguistic representations.

References

- Afouras, T., Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- Baevski, A., Schneider, S., and Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*, 2019.
- Baevski, A., Zhou, H., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.
- Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- Battenberg, E., Mariooryad, S., Stanton, D., Skerry-Ryan, R., Shannon, M., Kao, D., and Bagby, T. Effective use of variational embedding capacity in expressive end-to-end speech synthesis. *arXiv preprint arXiv:1906.03402*, 2019.
- Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- Boersma, P. Praat: doing phonetics by computer. <http://www.praat.org/>, 2006.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Bryant, G. A. Prosodic contrasts in ironic speech. *Discourse Processes*, 47(7):545–566, 2010. doi: 10.1080/01638530903531972. URL <https://doi.org/10.1080/01638530903531972>.
- Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., et al. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pp. 28–39. Springer, 2005.

- Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., and Poria, S. Towards multimodal sarcasm detection (an *_obviously_* perfect paper). *arXiv preprint arXiv:1906.01815*, 2019.
- de la Fuente Garcia, S., Ritchie, C., and Luz, S. Artificial intelligence, speech and language processing approaches to monitoring alzheimer’s disease: a systematic review, 2020.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. IEEE, 2017.
- Gratch, J., Artstein, R., Lucas, G., Stratou, G., Scherer, S., Nazarian, A., Wood, R., Boberg, J., DeVault, D., Marsella, S., et al. The distress analysis interview corpus of human and computer interviews. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pp. 3123–3128, 2014.
- Han, Y., Li, S., Cao, Y., Ma, Q., and Yoshikawa, M. Voice-indistinguishability: Protecting voiceprint in privacy-preserving speech data release. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2020.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., and Estève, Y. Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *International Conference on Speech and Computer*, pp. 198–208. Springer, 2018.
- Jain, U., Nathani, K., Ruban, N., Joseph Raj, A. N., Zhuang, Z., and G.V. Mahesh, V. Cubic svm classifier based feature extraction and emotion detection from speech signals. In *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, pp. 386–391, 2018. doi: 10.1109/SNSP.2018.00081.
- Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- Jansen, A., Plakal, M., Pandya, R., Ellis, D. P., Hershey, S., Liu, J., Moore, R. C., and Saurous, R. A. Unsupervised learning of semantic audio representations. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 126–130. IEEE, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., and Lee, H.-y. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6419–6423. IEEE, 2020.
- Liu, Z.-T., Wu, M., Cao, W.-H., Mao, J.-W., Xu, J.-P., and Tan, G.-Z. Speech emotion recognition based on feature selection and extreme learning machine decision tree. *Neurocomputing*, 273:271–280, 2018. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2017.07.050>. URL <https://www.sciencedirect.com/science/article/pii/S0925231217313565>.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Low, D. M., Bentley, K. H., and Ghosh, S. S. Automated assessment of psychiatric disorders using speech: A systematic review. *Laryngoscope Investigative Otolaryngology*, 5(1):96–116, 2020. doi: <https://doi.org/10.1002/lio2.354>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/lio2.354>.
- Luz, S., Haider, F., de la Fuente, S., Fromm, D., and MacWhinney, B. Alzheimer’s dementia recognition through spontaneous speech: The adress challenge. *arXiv preprint arXiv:2004.06833*, 2020.
- McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. Montreal forced aligner: Trainable text-speech alignment using kald. In *Interspeech*, volume 2017, pp. 498–502, 2017.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- Nyquist, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.
- Okada, S., Ohtake, Y., Nakano, Y. I., Hayashi, Y., Huang, H.-H., Takase, Y., and Nitta, K. Estimating communication skills using dialogue acts and nonverbal features in multiple discussion datasets. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*,

- ICMI '16, pp. 169–176, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345569. doi: 10.1145/2993148.2993154. URL <https://doi.org/10.1145/2993148.2993154>.
- Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- Oord, A. v. d., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.
- Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- Park, S., Shim, H. S., Chatterjee, M., Sagae, K., and Morency, L.-P. Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 50–57, 2014.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- Pompili, A., Rolland, T., and Abad, A. The inesc-id multimodal system for the adress 2020 challenge, 2020.
- Qian, K., Zhang, Y., Chang, S., Hasegawa-Johnson, M., and Cox, D. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pp. 7836–7846. PMLR, 2020.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- Ravanelli, M., Parcollet, T., Rouhe, A., Plantinga, P., Rastorgueva, E., Lugosch, L., Dawalatabad, N., Ju-Chieh, C., Heba, A., Grondin, F., Aris, W., Liao, C.-F., Cornell, S., Yeh, S.-L., Na, H., Gao, Y., Fu, S.-W., Subakan, C., De Mori, R., and Bengio, Y. Speechbrain. <https://github.com/speechbrain/speechbrain>, 2021.
- Razavi, A., Oord, A. v. d., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.
- Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
- Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.
- Shor, J., Jansen, A., Maor, R., Lang, O., Tuval, O., Quitry, F. d. C., Tagliasacchi, M., Shavitt, I., Emanuel, D., and Haviv, Y. Towards learning a universal non-semantic representation of speech. *arXiv preprint arXiv:2002.12764*, 2020.
- Siddiquie, B., Chisholm, D., and Divakaran, A. Exploiting multimodal affect and semantics to identify politically persuasive web videos. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI '15*, pp. 203–210, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2820732. URL <https://doi.org/10.1145/2818346.2820732>.
- Skerry-Ryan, R., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R., Clark, R., and Saurous, R. A. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pp. 4693–4702. PMLR, 2018.
- Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., and Khudanpur, S. Spoken language recognition using x-vectors. In *Odyssey*, pp. 105–111, 2018.
- Stehwien, S. and Vu, N. T. Prosodic event recognition using convolutional neural networks with context information. *arXiv preprint arXiv:1706.00741*, 2017.
- Takefuta, Y., Jancosek, E. G., and Brunt, M. A statistical analysis of melody curves in the intonation of american english. In *Proceedings of the 7th International Congress of Phonetic Sciences*, pp. 1035–1039. IPA Montreal, Canada, 1972.
- Tomashenko, N., Srivastava, B. M. L., Wang, X., Vincent, E., Nautsch, A., Yamagishi, J., Evans, N., Patino, J., Bonastre, J.-F., Noé, P.-G., et al. Introducing the voiceprivacy initiative. *arXiv preprint arXiv:2005.01387*, 2020.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- Voita, E. and Titov, I. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*, 2020.

- Wan, T., Kenter, V., Chan, C.-A., Clark, R., and Vit, J. Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pp. 3331–3340. PMLR, 2019.
- Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.
- Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pp. 5180–5189. PMLR, 2018.
- Woodland, J. and Voyer, D. Context and intonation in the perception of sarcasm. *Metaphor and Symbol*, 26(3):227–239, 2011. doi: 10.1080/10926488.2011.583197. URL <https://doi.org/10.1080/10926488.2011.583197>.
- Zadeh, A. B., Liang, P. P., Poria, S., Cambria, E., and Morency, L.-P. 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)*, pp. 2236–2246, 2018.
- Zhang, Y.-J., Pan, S., He, L., and Ling, Z.-H. Learning latent representations for style control and transfer in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6945–6949. IEEE, 2019.