Interpreting intermediate convolutional layers of CNNs trained on raw speech

Gašper Beguš and Alan Zhou

***Abstract*—This paper presents a technique to interpret and visualize intermediate layers in CNNs trained on raw speech data in an unsupervised manner. We show that averaging over feature maps after ReLU activation in each convolutional layer yields interpretable time-series data. The proposed technique enables acoustic analysis of intermediate convolutional layers. To uncover how meaningful representation in speech gets encoded in intermediate layers of CNNs, we manipulate individual latent variables to marginal levels outside of the training range. We train and probe internal representations on two models — a bare WaveGAN architecture and a ciwGAN extension which forces the Generator to output informative data and results in emergence of linguistically meaningful representations. Interpretation and visualization is performed for three basic acoustic properties of speech: periodic vibration (corresponding to vowels), aperiodic noise vibration (corresponding to fricatives), and silence (corresponding to stops). We also argue that the proposed technique allows acoustic analysis of intermediate layers that parallels the acoustic analysis of human speech data: we can extract F0, intensity, duration, formants, and other acoustic properties from intermediate layers in order to test where and how CNNs encode various types of information. The models are trained on two speech processes with different degrees of complexity: a simple presence of [s] and a computationally complex presence of reduplication (copied material). Observing the causal effect between interpolation and the resulting changes in intermediate layers can reveal how individual variables get transformed into spikes in activation in intermediate layers. Using the proposed technique, we can analyze how linguistically meaningful units in speech get encoded in different convolutional layers.**

***Index Terms*—convolutional neural networks, interpretability, speech, GANs**

I. Introduction

How deep convolutional neural networks learn their internal representations is one of the central questions in machine learning. The vast majority of work on this topic is centered on the visual domain. Here, we propose a technique to visualize and interpret the intermediate layers of deep convolutional neural networks trained on speech in an unsupervised manner. There are several advantages of interpreting intermediate layers in CNNs that are trained on speech over those trained on visual data.

First, humans process speech by discretizing the continuous physical properties of sound into discrete mental representations called phonemes. A long tradition of scientific study of phonetics and phonology ([1], [2]) has resulted in a relatively good understanding of how humans represent continuous properties in speech with such discrete phonemes. The main objective of CNNs trained on speech is similar to the discretization process in human phonology: the networks are trained to represent raw continuous data with discrete labeled categories. Speech data, combined with the study of human phonology, thus provides a highly interpretable testing ground for probing how learned representations emerge in deep neural networks, as well as in humans acquiring language.

Second, speech data contains multiple local and non-local dependencies with different degrees of computational complexity that are well-documented and well-understood. For example, changing or adding a single sound to a word can result in a change in meaning. For example, the English word *pit* [ˈp^{h}ɪt] has a very different meaning from *spit* [ˈspɪt]. Two processes occur here. First, the addition of the sound [s] changes the meaning of the word. Second, the stop consonant is produced with aspiration (puff of air marked by ^{h}) in the first word with no preceding [s], but without aspiration in the second word with preceding [s]. This contextually conditioned complementary distribution (between [p^h] and [p]) is computationally simple, but this is not true for all processes in human speech. For example, many natural languages feature an identity-based process called reduplication which requires phonological material to be copied from the output. A reduplicated form of the base [para] is [papara], where the first consonant and the first vowel [pa] in the base [para] are repeated (copied), which results in [papara]. Reduplication is computationally challenging [3], [4], because learners need to copy phonological material from the base: [pa] is the prefix only for the bases starting with [pa] such as [para]. For other bases (such as [tara] or [mura]), the reduplication morphemes that fulfill exactly the same function are substantially different: [ta] and [mu]. We can use these well-understood dependencies with different degrees of computational complexity to test what internal representations are learned from raw continuous data by CNNs and how they are learned, as well as which acoustic properties get encoded at which convolutional layer.

Finally, an advantage of interpreting CNNs trained on speech is that behavioral and acquisitional data is easy to obtain for speech. We can directly compare developmental stages in child language acquisition with stages of CNNs trained on speech [5], or use the same data to train human subjects and CNNs [6]. Outputs of the proposed technique can also be directly compared with human neuroimaging data, which contains time-series data of electrical activity similar to our outputs, obtained by recording different parts of the brain with various neuroimaging techniques.

Department of Linguistics, University of California, Berkeley. Emails: begus@berkeley.edu, azhou314@berkeley.edu

II. Prior work

Visualizing convolutional layers is performed primarily on models trained on visual data [7], with considerably less work focused on the visualization of convolutional layers of the models trained on speech [8], [9], [10], [11]. Work on unsupervised models such as GANs has primarily been carried out on image data, and has been successful in identifying relationships in the latent space [12], as well as intermediate representations of various generated classes [13]. These approaches often leverage techniques specific to the visual domain, such as attribute prediction and image segmentation.

Substantially less work exist on interpreting convolutional layers trained on speech, the majority of which operates on supervised models. The majority of proposals focus on interpreting and visualizing filters. The SincNet proposal [14] visualizes filters, and by imposing restrictions on filters, achieves better performance on an ASR task compared to unrestricted CNNs. Huang et al. [8] likewise focus on visualizing filters of convolutional layers from supervised models trained for ASR tasks. In [15], [16], [9], several visualization techniques are applied and topographic filter maps are visualized from supervised ASR models trained on spectrograms. The proposed techniques can highlight important regions for ASR tasks in CNNs. [17] analyzes activations in deep neural networks and correlates them with fMRI data.

Palaz et al., Muckenhirn et al., and Golik et al. ([18], [19], [20], [21], [22], [23]) also analyze learned filters at different convolutional layers. In [23], Muckenhirn et al. analyze filters of CNN models for ASR tasks, but trained on raw waveforms. They also visualize estimated F0 contours based on filters in the first convolutional layer [23]. Analysis of the filters can, for example, reveal which frequency bands various filters target. This can in turn reveal what types of acoustic data are encoded at which convolutional layers. However, the proposed techniques yield less directly interpretable outputs. For example, this technique does not allow a directly analysis of waveforms from individual convolutional layer that directly correspond to some phonetic element in the final output layer.

In [10], Muckenhirn et al. propose a gradient-based visualization technique for CNNs trained on raw waveforms (based on [24]) which yields relevance maps from the input signal that can be acoustically analyzed (a similar proposal that uses relevance maps is in [11]). Their models are trained on supervised tasks: phone or speaker identification. Similar to our technique, their proposal enables analysis of acoustic properties (such as formant values and F0) in CNNs on a time-series data. Their method, however, does not focus on analyzing which acoustic information is encoded at what layer and do not directly test effects of individual latent variables on convolutional layers. Additionally, they focus on spectral analyses as they argue that "[v]isualization in the time domain does not bring much insights into what important characteristics are extracted by the network because the results are difficult to interpret, as we do not have any visual cues as in the case of images". This paper argues that averaged ReLU activations of feature maps combined with manipulation and interpolation of individual linguistically meaningful latent variables yield highly interpretable time-series data.

Here, we propose a different approach from the existing proposals outlined above. Rather than analyzing convolutional layers in a supervised model or analyzing filters, our proposal focuses on the activations of intermediate transpose convolutions of a Generator network that was trained on speech in a GAN framework. Whereas traditional convolutions are usually used to downsample preexisting data into lower-dimensional representations, transpose convolutions work in reverse, upsampling from a low-dimensional latent representation in order to generate new data. This causes some key differences in the structure of our intermediate layers, with the highest-level representations appearing in the deepest layers of the network.

Our proposal brings several aspects that facilitate the interpretability of the activations of these intermediate convolutional layers: (i) manipulating and interpolating individual latent variables well beyond training range, (ii) averaging over feature maps to get intermediate time-series data, (iii) training in an unsupervised manner, and (iv) operating directly with time-series data.[1] Below we outline why each of these aspects is important. One of the main difficulties with interpreting convolutional layers in supervised ASR models is that it is not trivial to elicit or amplify activations given that the network takes raw data as inputs and outputs some classification. We propose an interpretable alternative by manipulating and interpolating individual latent variables that have linguistic meaning to values outside of training range in the Generator network. The generative and unsupervised aspect of the GAN framework make this technique possible: the Generator does not take raw data as inputs, but rather generates data. This means we can manipulate the latent space and observe causal effects of individual meaningful variables on intermediate layers.

Here we build on a proposal [5] that individual latent variables can be manipulated to marginal levels well outside the training range and that linear interpolation can reveal the causal relationship between individual variables and meaningful linguistic representations. The majority of proposals on CNN interpretability, known to the authors, do not manipulate individual latent variables. We claim that the main advantage of this approach is precisely interpretability: we can observe how individual variables with some linguistic function get transformed throughout the convolutional layers while keeping the rest of the latent space z constant.

We also interpret and visualize intermediate convolutional layers in a fully unsupervised manner — in the GAN framework. The majority of ASR models using CNNs are supervised. The advantage of interpreting intermediate layers on unsupervised models is that the final reduced representation layer is not trained on a classification problem with a softmax function, but is connected to a uniformly distributed random variables (or a combination of binary and uniformly distributed random variables) that get transformed to data in the final layer. This means that we can analyze self-organization of meaningful representations in intermediate convolutional lay-

[1] Other propsals also operate with raw waveforms (see above).

ers and directly observe effects of individual variables in the latent space on intermediate representations.

The same technique can also be applied to variational autoencoders (VAEs) trained on speech ([25], [26], [27], [28], [29], [30], [31]), but GANs are chosen because they are unsupervised not only in the encoding task, but also in the generative task and as such even more suitable for generating novel outputs. Unlike in VAEs, the generator of a GAN never directly accesses the training data. In other words, the generation aspect of VAEs is supervised — the network is trained on generating replicates of data and thus both its subparts, the encoder and the decoder, have direct access to the input data. In the GAN architecture, the generation aspect is fully unsupervised: the Generator needs to learn to generate data from noise without directly accessing the training data.

Finally, the output of the proposed technique is directly interpretable time-series data. Our proposal requires no further processing of the outputs: the proposed technique results in time-series data from each convolutional layer that directly correspond to the waveform output in the final layer. Many proposals (with the notable exceptions outlined above) operate with spectrograms; the advantage of analyzing waveforms directly is that no information is lost during the transformation between spectrogram and waveform, and waveforms allow for any acoustic analysis of the intermediate convolutional layer and of the final output. Here, we extract four acoustic properties from intermediate layers (duration, intensity, F0, and formant values) and correlate them to the final output in order to test how and in which layers does the Generator network encode various acoustic properties. Finally, waveforms can be directly played to participants in potential psycholinguistic experimental applications. In other words, spectral analysis is always available to raw waveforms, but converting spectrograms into raw waveforms is more challenging.

III. Models

The interpretation and visualization of individual layers is performed on the Generator network in two models: WaveGAN [32] and ciwGAN [33]. WaveGAN is a single-dimensional transformation of the DCGAN architecture [34] used for audio data. The architecture includes the Generator and the Discriminator networks. The Generator takes 100 latent variables z uniformly distributed in the interval $(-1, 1)$ and transforms them into 16,384 data points constituting just over 1 s of audio file (sampled at 16 kHz) through five convolutional layers. All layers except for the last one are trained with ReLU activation (tanh in the last layer). The dimensions are summarized in Figure 1. The Discriminator network takes real and generated audio files (16,384 data points constituting audio file) and estimates the Wasserstein distance between real data and generated outputs (according to the proposal in [35] with gradient penalty [36]). The Generator is trained on minimizing this distance, while the Discriminator is trained on maximizing it.

A. Bare WaveGAN on a simple conditional distribution

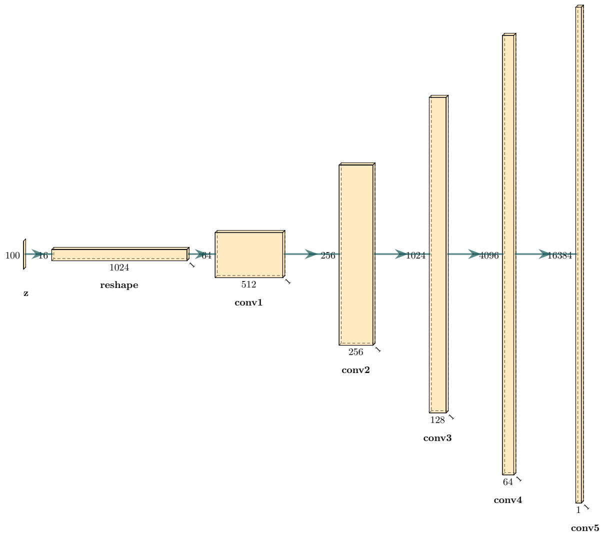

Fig. 1. The architecture of the Generator network with five one-dimensional convolutional layers as proposed in [32] and used for training in this paper. Filters are one-dimensional with the size of 25 [32].

First, we analyze how the three basic acoustic properties of speech are encoded in CNNs: periodic vibration corresponding to vowels, aperiodic noise corresponding to fricatives (such as [s]), and silence corresponding to the closure part of stop consonants. For this purpose, we perform an analysis on the Generator network trained in [5] on sliced sequences of the structure #sTV and #TV from TIMIT [37] (where T = /p, t, k/ V = vowel, # = word edge). Altogether 5,463 data points from TIMIT were used for training: 4,930 sequences of the structure #TV (such as [ˈp^{h}æ]) and 533 of the structure #sTV (e.g. [ˈspæ]). We used simplified training materials to facilitate interpretation of intermediate layers, but the visualization technique proposed here is scalable to more complex training data too. The network was trained for 12,255 steps (approximately 716 epochs) [5]. At this point, the network not only learns to output speech-like sequences (#TV and #sTV) that resemble training data and are acoustically analyzable, but also learns the simple conditional distribution in which aspiration is shortened if [s] is present in the output ([ˈp^{h}æ] vs. [ˈspæ]).

In [5], we propose a technique to identify those latent variables from z that correspond to some meaningful linguistic representation, such as presence of [s], using a combination of shrinkage techniques, linear and non-linear regression, and random forest. We argue that the Generator learns to represent presence of [s] with a subset of latent variables z. Crucially, we show that manipulating the variables chosen with the regression technique results in an almost one-to-one mapping between *individual* latent variables and presence of [s]. We perform several generative tests to confirm the link between individual latent variables and the presence of some linguistically meaningful unit, such as [s]. We propose that by manipulating individual variables to marginal levels well outside of training range (i.e. well outside the interval $(-1, 1)$) to values such as ±15, we can force [s] to surface in the output at near categorical levels [5], [33].

Manipulating individual latent variables to marginal values well outside of the training range to create a high occurrence of a desired linguistic unit is a crucial concept used in this paper. This technique reveals that the Generator learns to use the latent space as a discretized representation of linguistically meaningful units. For example, using regression techniques, we identify 7 variables z_i out of the 100 in the latent space that strongly correspond to presence of [s] in the output. The eleventh variable z_{11} is one such variable that strongly corresponds to presence of [s]. By setting z_{11} to -1 (within the training range), we get a modest increase of [s]-containing sequences in the output. By setting it to -15, 87% of outputs contain an [s]; by setting it to -25, there are 96% such outputs. Other linguistically meaningful units including morphological items (such as affixes) have been shown to be encoded with discretized representations. For example, in [6], we show that the network learns to represent a prefix of the shape V(N)- (where V = a vowel and N = a nasal consonant, e.g. ɔn-) in a highly discretized manner. The network is trained on prefixed and unprefixed forms (e.g. [ˈpʰɔɹɔ] and ɔm-ˈpʰɔɹɔ]). The analysis of the latent space reveals a substantial spike in regression estimates for a single variable corresponding to the presence of the prefix: x_{16}. When this variable is set to -4.5 (outside of training range), 100% of outputs contain a prefix; when it is set to the opposite 4.5, 99% of outputs lack a prefix [6]. Linear interpolation between the two marginal values results in a linear shift between prefixed and unprefixed forms.[6]

Similarly, we show that, in the model trained on #TV and #sTV sequences, interpolating z_{11} from marginal values results in a gradual reduction of frication noise in the output until [s] ceases from the output. The frication noise of [s] appears to be directly causally connected with z_{11}. Figure 13 shows how interpolating z_{11} from 5 (corresponding to absence of [s] in the output) to -15 (corresponding to presence of [s] in the output) results in the gradual appearance and then increase of frication noise in the generated output. In [5], it is shown that direct correlations between single latent variables and the amplitude of frication noise of [s] in the output operate across generated samples and persists even when the amplitude is measured proportionally to the vocalic amplitude. In sum, we argue that there is a causal relationship between individual latent variables identified with the proposed technique [5] and linguistically meaningful properties of the output.

B. CiwGAN on an identity-based pattern

The conditional allophonic distribution described above is computationally among the simplest processes in human languages. To test whether the technique for interpretation of intermediate layers extends to computationally more complex processes in language, we apply the technique to the ciwGAN model trained on an identity-based pattern (copying) called *reduplication*. The ciwGAN architecture differs from the bare WaveGAN in that it includes a Q-network [33]. The Generator takes as input categorical code variables c in addition to the latent variables z. The code variables constitute a one-hot vector. The Q-network is in structure identical to the Discriminator except in its final layer. The Q-network takes as input the Generator's outputs and estimates the latent code c used by the Generator. The Q-network and the Generator are trained to maximize the Q-network's success rates (the architecture is summarized in Figure 2). In other words, the proposed architecture forces the Generator to output informative data. For example, when the ciwGAN network is trained on words from TIMIT, the most informative way to encode unique information (e.g. a one-hot vector) into acoustic data is to associate each word with unique latent code c. Lexical learning thus emerges automatically from only the requirement that the Generator produce informative data in a completely unsupervised manner – lexical items are never labeled during training. Training thus results in a Generator that learns to output unique words for each latent code [33].

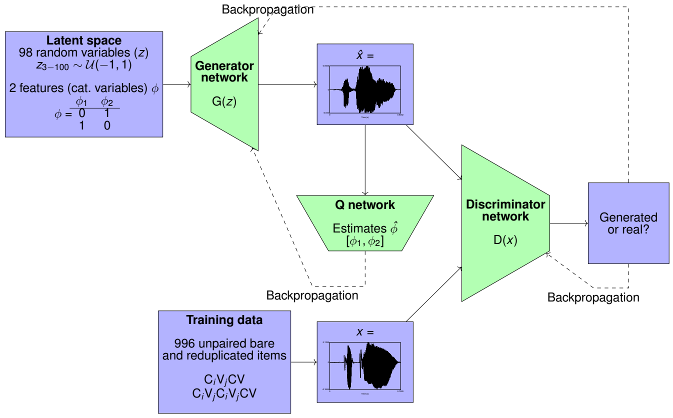

Fig. 2. The ciwGAN architecture as proposed in [33] used for interpreting intermediate layers in Section IV-B. Figure taken from [38].

The advantage of the ciwGAN architecture is that learning of linguistically meaningful units emerges from the requirement that the Generator outputs informative data. To test how learning of a highly complex process such as reduplication self-emerges in this architecture, we train the ciwGAN network (in [38]) with one-hot latent code of length 2 on 996 bare and reduplicated items (e.g. [ˈpʰɑli] and [pʌˈpʰɑli]). The model is trained for 15,920 steps (or approximately 5,114 epochs). The Generator learns to associate the latent code with reduplication: when latent code is set to marginal levels of 5 [5, 0], the Generator outputs 98% unreduplicated forms; when it is set to [0, 5], it outputs 87% reduplicated forms [38]. When the values are interpolated, the Generator gradually turns a bare unreduplicated form into a reduplicated form (e.g. from [ˈpʰiɹu] to [pəˈpʰiɹu] [38]). Figure 14 shows how manipulating categorical latent variable c_2 results in the gradual appearance of a reduplicated syllable in the output. The network also learns to extend the learned pattern to unobserved data and reduplicates forms with initial consonants that were withheld from training. For example, by simultaneously forcing reduplication and [s] in the output (setting the latent variables to marginal levels beyond training range), the network outputs [səˈsiji], although [səˈsiji] and all [s]-containing reduplicated forms were withheld from training data (the network only sees unreduplicated [s]-initial words such as [ˈsiji]). These results strongly suggest that the Generator learns to represent a linguistically meaningful and computationally highly complex process (reduplication or copying) with the latent codes in a fully unsupervised manner.

In [5], [33], [38], [6], we only analyze and interpret the endpoints of these models: the latent variables and the generated outputs. Here, we propose that intermediate convolutional layers can be interpreted using this technique as well. We argue that by manipulating individual latent variables to marginal values and interpolating from those marginal values, we can reveal how individual variables in the latent space z cause spikes in activations in intermediate layers and how linguistically meaningful units get represented throughout the convolutional layers.

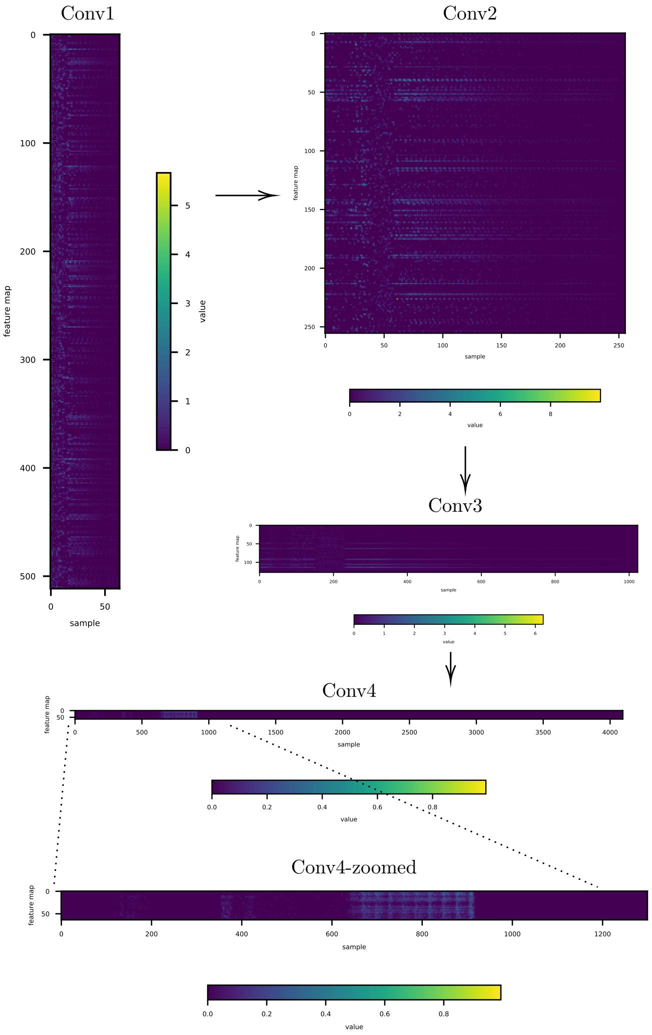

Fig. 3. Values of feature maps (concatenated on the y-axis) after ReLU activation in four convolutional layers for a uniformly distributed z-vector limited to the training interval $(-1, 1)$. The visualization illustrate dimensions of convolutional layers in the Generator network. The visualizations illustrate how activations in the previous layers result in a clearly analyzeable periodic vocalic structure in the fourth convolutional layer (Conv4 on the zoomed-in graph) that results in a vowel in the output.

IV. Interpretation

We propose that learned representations in the intermediate layers can be evaluated by combining two techniques: (i) averaging across feature maps in each layer after ReLU activation (Sections IV-A and IV-B) and (ii) manipulating individual z variables to marginal values well outside the training range (Section IV-C). To evaluate the causal relationship between individual latent variables and the convolutional layers, the z variables can be interpolated from marginal endpoints outside of the training range. The proposed technique reveals which features in the intermediate layers get activated when manipulating individual latent variables z and which linguistically meaningful variables (such as duration, F0, intensity, or formant structure) get encoded at which layers.

This approach also allows us to follow how interpolation of individual latent variables z (such as z_{11}) that correspond to some meaningful linguistic unit (such as presence of [s] or reduplication) affect individual feature maps in each convolutional layer (Figure IV-D).

A. Model 1: WaveGAN [32]

The Generator network is a five-layer 1D deep convolutional network. The dimensions of the five convolutions are $512 \times 64 \times 1$, $256 \times 256 \times 1$, $128 \times 1024 \times 1$, and $64 \times 4096 \times 1$. The final layer (with tanh activation) has a dimension of $16384 \times 1 \times 1$ that constitutes just over 1 second of audio waveform (16 kHz sampling). The architecture is summarized in Figure 1. Figure 3 plots values of each feature map (concatenated along the y-axis) for a z that is randomly distributed on the training interval $(-1, 1)$ across all variables. The visualization illustrates the structure of the Generator. At the fourth convolutional layer, a clear periodic structure of the vocalic part is visible. The most common technique of visualizing CNNs — a simple concatenation of feature maps — does not provide the most interpretable results in speech beyond these basic observations.

We propose that averaging across all feature maps results in highly interpretable time-series data. Figure 4 plots the third (Conv3) and fourth (Conv4) convolutional layers, averaged across all feature maps after ReLU activation along with the corresponding waveform output that can be transcribed as involving a fricative [s], a stop, and a vowel (#sTV). Overlaying the last two convolutional layers with the generated output reveals that the fourth convolutional layer includes information for all three major acoustic properties of the output: we observe a period of aperiodic vibration corresponding to the frication noise (in [s]), a period of silence corresponding to the closure portion of the consonant (T) and a clear periodic vibration corresponding to the vowel (V). The timing of these constituents in Conv4 aligns completely with the generated output.

The fourth layer (Conv4) carries both the fundamental frequency (F0) and formant structure information in the vocalic part of the input. Figure 4 (middle) clearly shows that the averaged fourth convolution after ReLU contains periodic vibration with the fundamental frequency that matches the output and higher-frequency vibration that corresponds to the formant structure in the output. There appears to be also the amplitude/intensity information in the fourth layer — Conv4 closely traces the actual output in the final output layer.

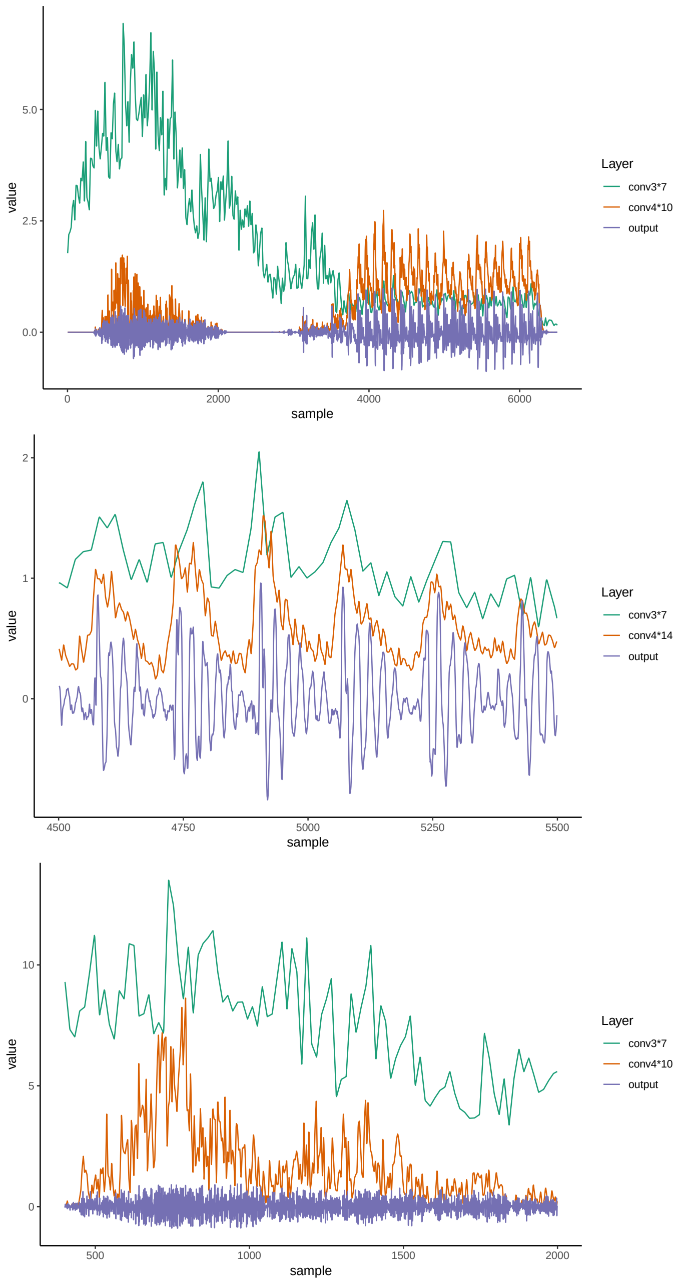

Fig. 4. All feature maps averaged after ReLU activation after the third convolutional layer (conv3; green), fourth convolutional layer (conv4; dark orange) and the generated output (output; purple). (top) A generated output when $z_{11} = -5$ featuring a period of frication [s], a period of silence (of a stop consonant), and a vocalic period. To overlay the two convolutional layers on top of the output, they are multiplied by 7 and 10, respectively. (middle) A period of vocalic periodic vibration with the same latent space values as above, but z_{11} set at -1 and conv3 and conv4 multiplied by 7 and 14, respectively, to overlay the convolutional layers on top of the output. (bottom) A period of frication (in [s]) with the same latent space values as above, but z_{11} set at -11 and conv3 and conv4 multiplied by 7 and 10, respectively, to overlay the convolutional layers on top of the output.

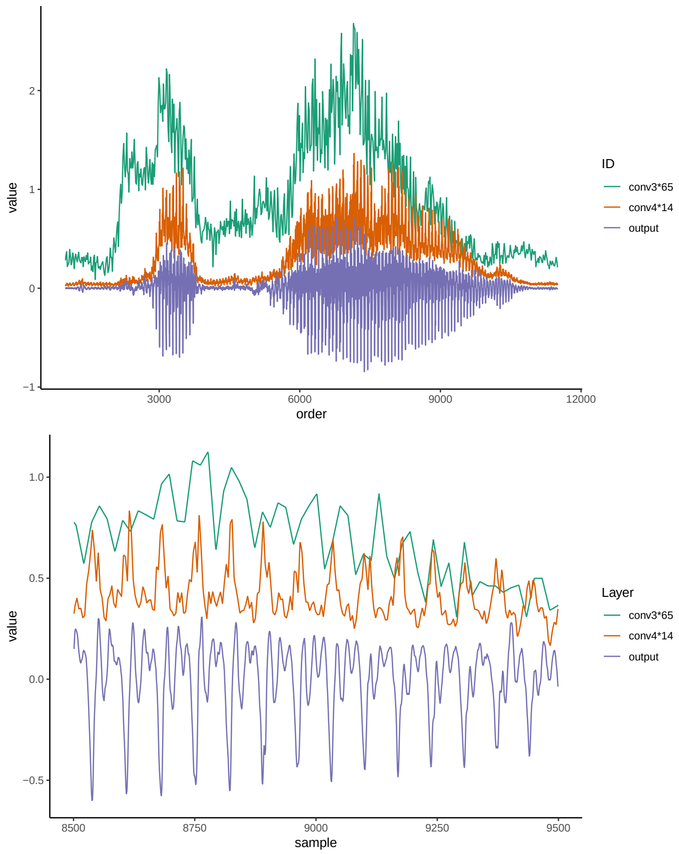

Fig. 5. All feature maps averaged after ReLU activation after the third convolutional layer (conv3; green), fourth convolutional layer (conv4; dark orange) and the generated output (output; purple). (top) A generated output when $c_2 = 1$. To overlay the two convolutional layers on top of the output, they are multiplied by 65 and 14, respectively. (bottom) Zoomed-in enerated output when $c_2 = 1$.

To quantify these observations, we randomly generate 25 outputs from the bare GAN model trained after 12,255 steps on #TV and #sTV sequences and convert outputs from intermediate layers to waveforms ready for acoustic analysis.[2] We manually annotate the vocalic period in the final output and perform acoustic analysis of the outputs in the third and fourth convolutional layers (Conv3 and Conv4).

1) Duration: We manually annotate periodic vibration in the fourth convolutional layer and compare vowel durations of the 25 generated outputs between the final output and the fourth convolutional layer. The vocalic durations are easily identifiable in Conv4 and nearly identical to the vocalic duration in the final output. Durations from the two layers fit to a linear model reveal a high degree of correlation ($\beta = 0.96, t = 30.31, p < 0.0001$) with adjusted $R^2 = 0.97$. Figure 7 illustrates the correlation. In the averaged Conv3-output, the difference between the periodic vibration characteristic of vowels and other acoustic properties, such as silence (characteristic of stops) or frication noise (characteristic of fricatives and aspiration), are not clearly visible. Figure 6 plots three averaged Conv3 layers and the vocalic period annotated from the corresponding output of the final layer. Figure 13 additionally shows that in Conv3, the period of

[2] As the intermediate layers are all positive, we clip all values greater than 1 to be equal to 1 in the waveform outputs. We then treat the signal as a float32 signal and convert it to a .wav file. We also upsample the intermediate layers to 16 kHz sampling with linear interpolation.

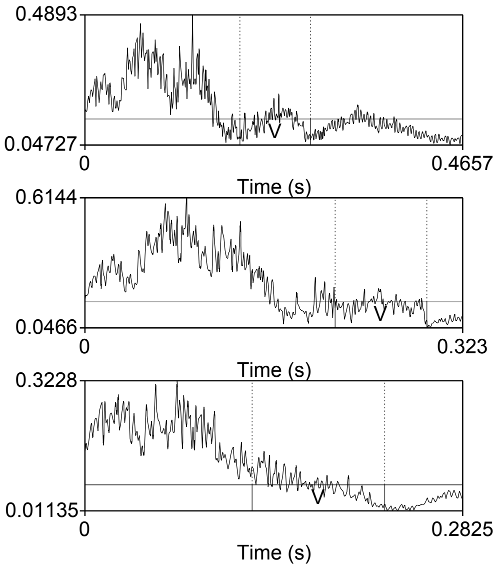

Fig. 6. Averaged outputs in the third convolutional layer (Conv3) after ReLU activation with vocalic periods (labeled with V) annotated from the final output.

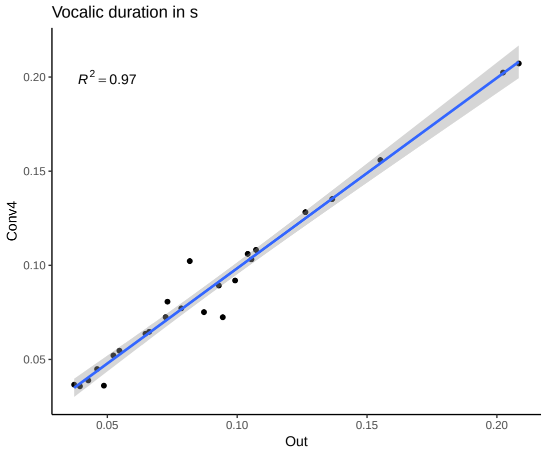

Fig. 7. Correlation in duration (in seconds) of vocalic periods between the generated output in the final layer (Out on x-axis) and fourth convolutional layer (Conv4 on y-axis).

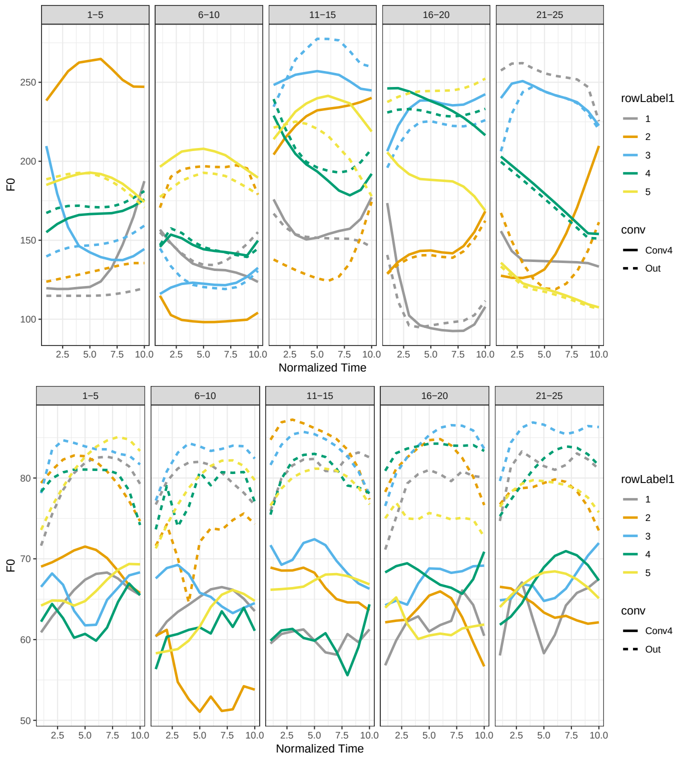

Fig. 8. (top) F0 values in normalized time (10 intervals) in 25 randomly generated outputs for the final output (Out) and fourth convolutional layer (Conv4). The values were extracted using the Praat software [39] with a script by Xu [40]. The window for F0 range was set to 60-300Hz for the analysis. (bottom) Intensity values in normalized time (10 intervals) in 25 randomly generated outputs for the final output (Out) and fourth convolutional layer (Conv4). The values were obtained as described for F0 (the minimum pitch for intensity is 100 Hz).

silence corresponding to the stop consonant between the [s] and the vowel in #sTV sequences is not encoded as strongly as in Conv4 when the values of individual latent variables are interpolated.

Based on these results, we can conclude that vocalic duration and periods of silence corresponding to stop closure is most strongly encoded in the fourth convolutional layer (Conv4) in the model trained on #TV and #sTV sequences.

2) F0: To test how the Generator encodes the fundamental frequency (F0), we extract F0 values from the annotated vocalic period in the 25 randomly generated outputs.[3] The Conv4 outputs are noisy and limited to positive values, which is why extraction of F0 can be challenging. The range of F0 is set to 60–300 Hz for the analysis. Figure 8 shows the 50 extracted values (25 for each layer). Several F0 trajectories are almost identical between the final layer and Conv4. A correlation test of concatenated values between the two layers (Conv4 and output reveals a substantial correlation with $R^2 = 0.53$ (Pearson's product-moment correlation).

Figure 4 suggests that pitch is likely also encoded in Conv3. Conv3 layer shows peaks that correspond to vocalic periodic vibration. However, with the relatively weak signal, F0 contours are difficult to extract from the Conv3 of a model that is trained with relatively few steps. For further discussion, see Section IV-B1 on F0 in Conv3 in a model trained with more steps.

3) Intensity: To test whether and how intensity is encoded in Conv4 (as observed in the qualitative analysis in Figure 4), we extract intensity values from annotated vocalic periods (using the the script by Xu [40] in Praat [39] with 100 Hz minimum pitch). Figure 8 illustrates that the intensity values of Conv4 are lower compared to the final output, but there is a correlation between concatenated values of intensity in the two layers: $R^2 = 0.62$. Lower overall values of the intensity levels are expected as the Conv4 layer only includes positive values and there is no reason for the network to match intensity values in absolute terms across the layers. The measure of interest

[3] For the purpose of analyzing F0 and intensity, we use annotations of the vocalic period from the final output (Out) also for the analysis of F0 and intensity in the third and fourth convolutional layers (Conv3 and Conv4).

here is independent of absolute values: we are interested in intensity trajectories, which do show a substantial correlation between Conv4 and the final output.

Since intensity is easier to extract from noisy data (compared to F0), we also correlate intensity levels between the third convolutional layer (Conv3) and the final output. Because vocalic period is not clearly encoded in Conv3, we use annotations of the vocalic period from the final output. There is a modest correlation in intensity values between Conv3 ant the final output: $R^2 = 0.39$. Figure 13 also suggests that intensity (or amplitude envelope) is encoded in Conv4, Conv3, and perhaps even in Conv2 when individual latent variables are manipulated to marginal values.

B. Model 2: CiwGAN [33]

To test which acoustic properties are encoded most strongly in which convolutional layer in the ciwGAN model trained on an identity-based pattern – reduplication, we generate 30 random outputs, 15 each for the two values of the code variables ([0, 1] and [1, 0]). To evaluate acoustic properties over longer time frames of periodic vibration, we extract F0 and intensity values over the entire periodic vibration of an output (all voiced sounds). For example, in an output transcribed as [ˈbɑli], the F0 and intensity values are extracted from all sounds, because they are all voiced.[4]

1) F0: F0 values are challenging to extract given that the averaged values after ReLU activations are only positive. Outputs from the ciwGAN model suggest that F0 is already encoded in the fourth convolutional layer, similarly to what is suggested in the bare WaveGAN model. The extracted F0 values often suffer from doubling and halfing errors, but there is still a correlation between F0 in the output and in the fourth convolutional layer (Conv4): $R^2 = 0.55$.

The ciwGAN model also suggests that the F0 is at least partly encoded already in the third convolutional layer (Conv3). Figure 9 plots all extracted F0 values from the final output and the third convolutional layer. There is a moderate correlation in F0 between the averaged Conv3 layer and the final output ($R^2 = 0.40$), suggesting that the F0 values are at least in part encoded already in the third convolutional layer.

Convolutions higher than the third layer (Conv3) cannot encode F0 in a non-abstract way: with a dimension of only 256, its Nyquist frequency is only 128 Hz. It is of course possible that different F0 values and trajectories are encoded in an abstract reduced representation in higher convolutions as well as in the latent space.

2) Intensity: Intensity appears to be strongly encoded both at the fourth and third convolutional layers. Figure 10 plots all 38 intensity trajectories for periodic vibration in the output and Conv3 and Conv4. Contrary to the analysis in Section IV-A3, intensity values in this model span not only a single vowel but often multiple vowels and voiced consonants (both sonorants and stops). Correlation between the concatenated final output values and averaged Conv4 values are high: $R^2 = 0.82$. The correlation between the output and averaged values from the third convolutional layer is slightly smaller, but nevertheless relatively high: $R^2 = 0.72$.

3) Formants: To test how formants are encoded in the Generator network, we extract the first and second formant values F1 and F2 (using script FormantPro by Xu and Gao [41] in Praat [39]).

The relationship in formant values between the output and Conv4 is complex. First, formants are relatively challenging to estimate, even in clean human acoustic data, let alone in generated data or in intermediate convolutional layers. Second, while the fourth convolutional layer clearly features formant structure, the relationship between Conv4 and the final output is not straightforward. Figure 11 illustrates this relationship. The spectrogram of the output [təˈtʰɑjə] in Conv4 reveals a clear formant structure (Figure 11) but the actual formant values only partially overlap with the final output layer.

To quantify this observation, we analyze formant values of the 38 periods with vocalic vibrations in normalized time and test the correlation between the fourth convolutional layer and the final output. The strongest correlation between the final output and the fourth layer appears to be in values of the second formant (F2): $R^2 = 0.40$. Figure 12 illustrates this correlation. In some outputs in the fourth convolutional layer (Conv4), F2 values match the final output layer both in the absolute values and in trajectories, but there also exist substantial deviations between the two layers. F2 is in a few cases already above the Nyquist frequency for Conv4 (2,048 Hz). F1, on the other hand, does not appear to be faithfully encoded in Conv4: a correlation test between the output and Conv4 suggest a negative correlation for F1 ($R^2 = -0.38$).

C. Interpolation

Results of the quantitative acoustic analysis of intermediate convolutional layers in Section IV-A and IV-B reveal how and where the Generator encodes different acoustic properties. To interpret how linguistically meaningful representations in the latent space translate into spikes in activation in the intermediate layers, we use the proposal in Beguš [5] and interpolate individual latent variables to marginal levels well outside the training range.

To test how linguistically meaningful representations are reflected in intermediate layers, we interpolate values of z_{11} in the bare GAN model and values of the latent code c_1 and c_2 in the ciwGAN model. We generate outputs by interpolating z_{11} from -15 to 5 (in increments of 2) which results in 11 outputs per each convolutional layer (55 total). One such set is chosen for visualization. The effects of interpolation are similar across all sets. All other 99 latent variables remain constant across all outputs. The set with 11 interpolated values across the five convolutional layers is plotted in Figure 13. The final output layer illustrates how an output without [s] gradually transforms into an output with [s] as z_{11} is interpolated towards the negative values which represent the presence of [s].

The advantage of the technique proposed in [5] is that we can observe the causal effect of individual latent variables on the output at each convolutional layer by analyzing averaged

[4] For reduplicated outputs interrupted by a stop, we extract the values separately for each periodic vibration, which totals in 38 analyzed periods from 30 outputs.

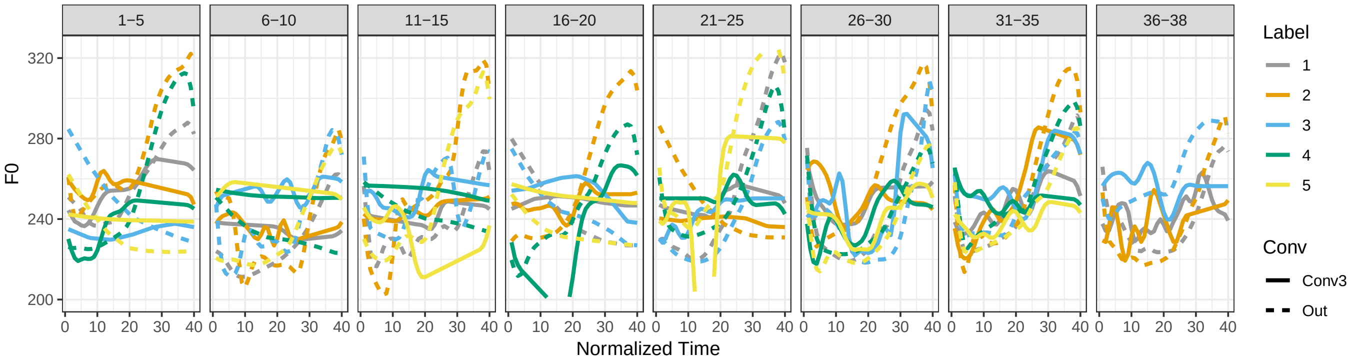

Fig. 9. F0 values in normalized time (40 intervals) in 30 randomly generated outputs (15 for each code; 38 vocalic periods total) for the final output (Out) and third convolutional layer (Conv3). The values were extracted using the Praat software [39] with a script by Xu [40]. The window for F0 range was set to 75-450 Hz for the analysis. Values below 250 Hz and above 325 are excluded from the plot.

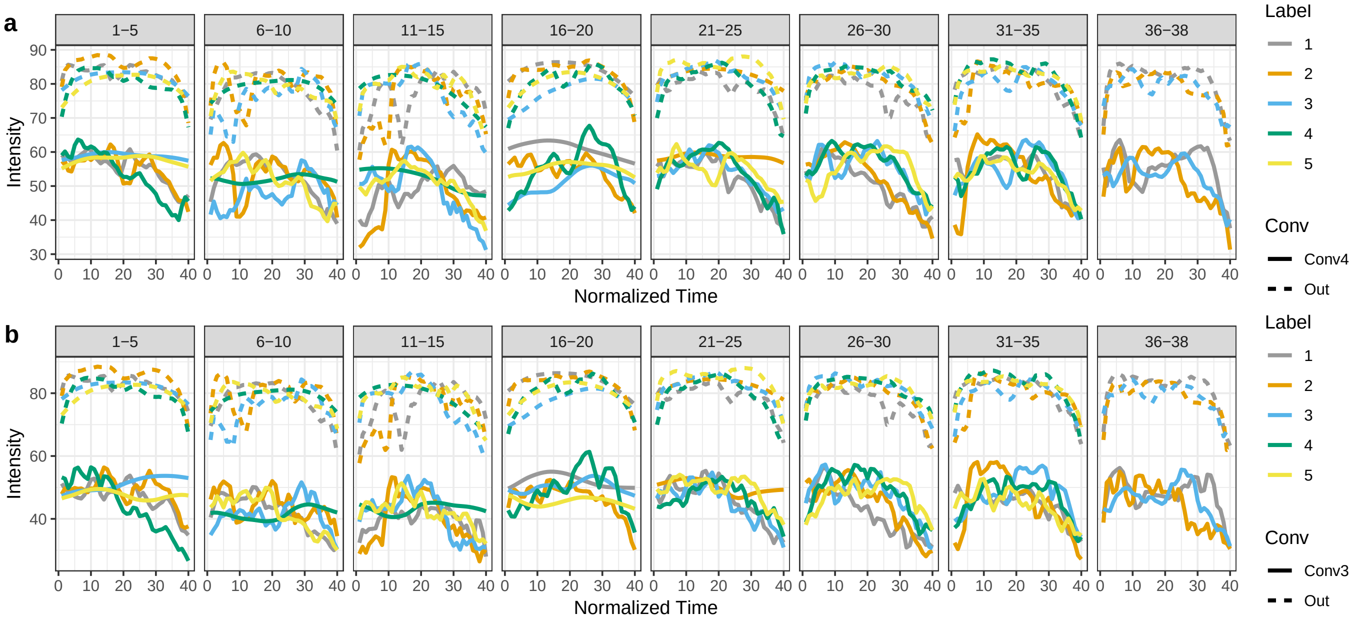

Fig. 10. Intensity values in normalized time (40 intervals) in 30 randomly generated outputs (15 for each code; 38 vocalic periods total) for the final output (Out) and fourth convolutional layer (Conv4) (in **(a)**) and third convolutional layer (Conv3) (in **(b)**). The values were obtained as described for F0 (the minimum pitch for intensity is 100 Hz).

ReLU activations. Figure 13 illustrates how the interpolation of z_{11} results in spikes of four values in the first convolutional layer. These four spikes increase as the values of z_{11} decrease, to the exclusion of other variables at this layer. It is likely the case that at the first layer, the discretized abstract representation of [s] in the latent space transforms into spikes of a subset of values. At this point, the transformation is still highly abstract. In the second convolutional layer (Conv2), the spikes transform into a more detailed representation of what corresponds to frication noise of [s] in the final output layer. The differentiation between the frication noise and periodic vocalic vibration becomes clearer in the third convolutional layer. The increasing amplitude of the period corresponding to frication noise (compared to the vocalic period) as the values of z_{11} approach -15 suggests that the four spikes in values from Conv1 transform into precursors of frication noise and that interpolation of the individual latent variable z_{11} representing [s] amplifies only the frication period to the exclusion of vocalic period throughout the four layers and the final output. There is thus a causal relationship between z_{11} and precursors of the frication noise at each convolutional layer. Visualization of the interpolations in the fourth layer (Conv4) also suggests that this layer encodes all major acoustic properties: frication noise, period of silence, and vocalic vibration as well as F0 and intensity of the periodic vocalic vibration.

To interpret latent code interpolation in the ciwGAN model trained on reduplication, we create a similar set: we manipulate the latent code from [0, 0] to [0, 2] in increments of 0.25, thus creating 9 outputs per convolutional layer (45 total). One such set is chosen for visualization, but the effects of interpolation are similar across all sets. All other 98 latent variables z remain constant across all outputs.

Interpretation of interpolated intermediate layers in the

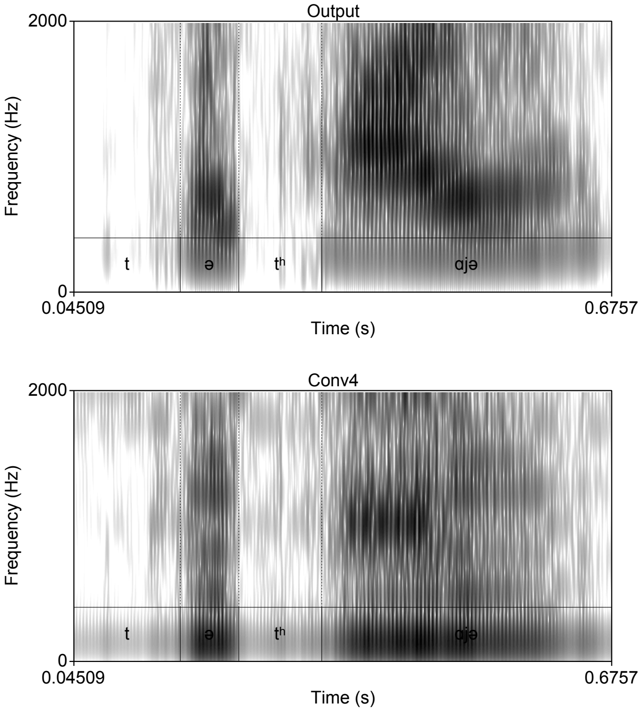

Fig. 11. Spectrograms (0-2000 Hz) of (top) the final generated output of a reduplicated form [taˈtʰɑjə] (from the ciwGAN architecture when $c_1 = 0$ and $c_2 = 1$) and (bottom) of the same reduplicated form with $c_1 = 0$ and $c_2 = 1$, but from the fourth convolutional layer (averaged across the feature maps after ReLU activation).

ciwGAN model is more complex because the phonological process the model is trained on — reduplication (or copying) — is computationally highly complex. The first convolutional layer shows less discretized representations than in the #sTV model. Interpolation from [0, 0] to [0, 2] (corresponding to presence of reduplication) seems to activate a few spikes for the main vowel and reduplicant vowel, but less categorically so than in the #sTV model. Intensity (or acoustic envelope) appears to be encoded through all the convolutional layers. Averaged ReLU activations with interpolated codes in Figure 13 suggest that the latent code representing a computationally complex process results in the formation of two vocalic periods, interrupted by an identical element as the one in the base (the copying principle). Visualizations also show that the period of silence (or reduced amplitude) corresponding to stop closure is encoded well into the third convolutional layer.

D. *Individual feature maps and interpolation*

1) Individual feature maps: In addition to the averaged feature maps at each layer, we also attempt to identify how linguistically meaningful features might be encoded separately in individual feature maps. Individual feature maps tend to be highly sporadic, with the same feature map possibly encoding different features even when the generated output is similar. However, there do exist some broad patterns across different generated outputs.

To identify these patterns for specific features, we generate a large number of different outputs, half of which we manipulate the latent space so that the feature of interest is present, and half of which we manipulate so that the feature is absent. Then, we average the activations together, and perform clustering to obtain broad categories. We manipulate the generated outputs because the raw outputs of the generator may have a highly imbalanced distribution of features.

We perform this analysis on the fourth convolutional layer of the WaveGAN model (Section III-A), generating 1000 total outputs, 500 of which have $z11$ set to -15, and 500 of which have $z11$ set to 15. We then perform spectral clustering, using the rbf kernel with a kernel coefficent of 1×10^{-10} to construct the affinity matrix, and clustering using k-means where $k = 2$. The results are shown in Figure 15.

We see two broad distributions of activations: one in which there is a spike of activation near the beginning of the waveform and relatively low activation afterwards, and another in which we see two additional spikes afterwards. We interpret the large single spike in the former category to correspond with the presence of a [s] frication, and determine these feature maps to encode almost exclusively for the presence of [s]. The latter category we take to also encode for [s], but which in addition is responsible for the rest of the #sTV sequence. Indeed, when we average these clusters separately for particular examples of generated words with and without the [s] frication in Figure 15, we see that the first cluster is activated very weakly compared to the second in the absence of [s]. In the presence of [s], we see activations from both clusters in the area corresponding to the [s]-frication, but only very small activations from the first cluster in the rest of the #sTV sequence.

2) Interpolation: . We can also analyze and interpret individual feature maps by interpolating individual latent variables with linguistically meaningful representations. Figure 17 illustrates four "raw" feature maps with interpolated values of z_{11} (in blue) and their corresponding final output layer (in gray). The four feature maps were chosen as those in which the distance between the feature map when z_{11} is -15 and each corresponding feature map when z_{11} is interpolated is smallest (according to cosine distance).

Individual feature maps show several parallels to the averaged values discussed in Section IV-C. By manipulating individual variables with linguistically meaningful representations (such as z_{11}), we can follow the causal effects of those variables on individual feature maps. Figure 17 illustrates that individual feature maps transform marginal z_{11} values into spikes in few values in Conv1. At Conv3, the z_{11} transforms into a less abstract representation of frication noise that substantially increases in amplitude as the values of z_{11} approach -15. At Conv4, we see differentiation into periods of frication noise, silence, and periodic vocalic vibration. Again, interpolation results in increased amplitude of the frication noise.

Visualization of individual feature maps combined with interpolation of individual linguistically meaningful latent variables thus allows us to explore whether individual feature maps separately encode different phonetic properties (e.g. frication noise, silence, or periodic vocalic vibration).

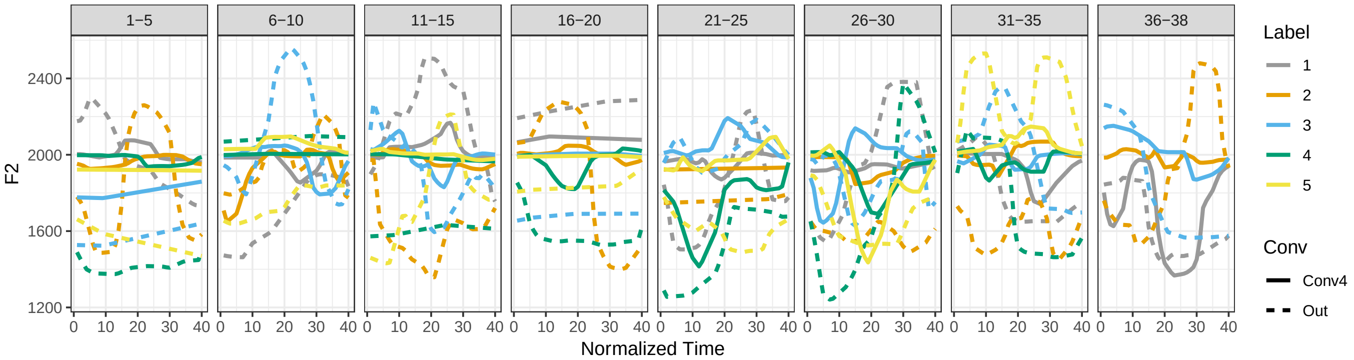

Fig. 12. F2 values in normalized time (40 intervals) in 30 randomly generated outputs (15 for each code; 38 vocalic periods total) in the final output (Out) and fourth convolutional layer (Conv4). The maximum formant value was set to 4500 Hz (with 4 as the maximum number of formants and a window of 25ms).

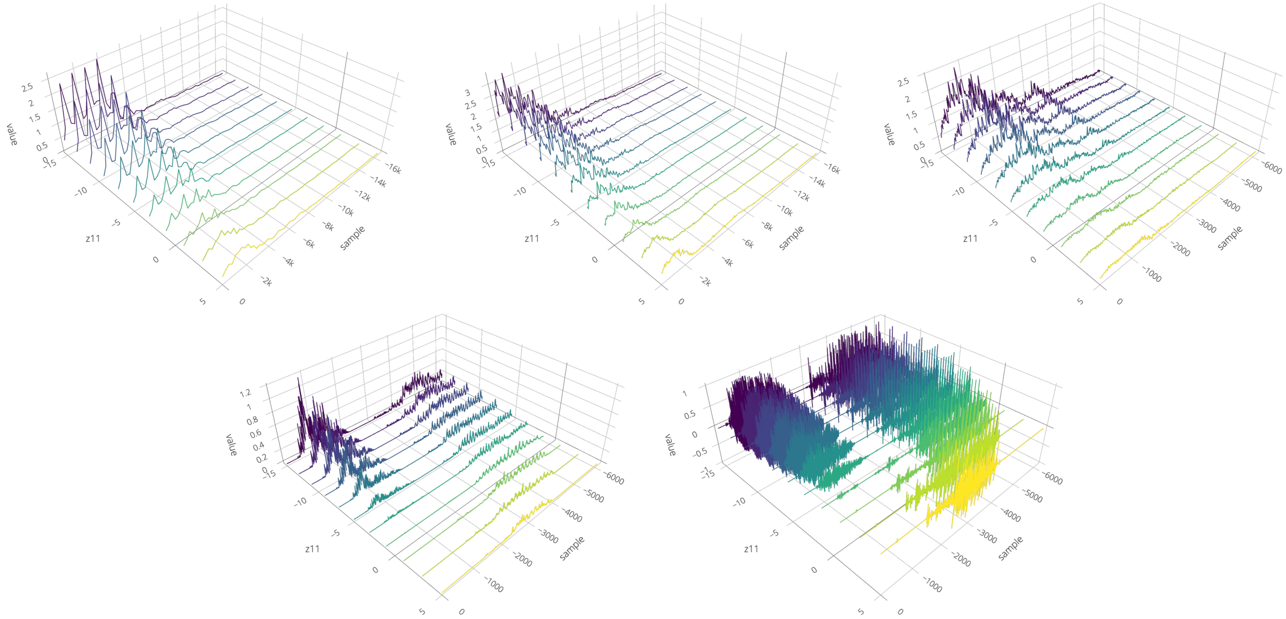

Fig. 13. Averaged values across feature maps after ReLU activation in the first (top left), second (top middle), third (top right), fourth (bottom left) convolutional layer, and the final waveform output (bottom right). For each convolutional layer, the graph represents 11 averaged values after ReLU activation where z_{11} is linearly interpolated from -15 to 5 (in increments of 2) while all other 99 latent z variables are held constant and limited to the training interval (-1,1) with uniform distribution. All outputs except in the final layer are upsampled with linear interpolation to total 16,384 samples (y-axis) to match the audio waveform output. Representation of the third, fourth, and final layer were cut off at 6100th sample because higher samples featured mostly silence. The figures illustrate how interpolating z_{11} from 5 to -15 results in appearance of sound [s] in the final output and how representation of [s] is encoded across the layers.

V. Discussion

This paper proposes a technique to interpret and visualize outputs at intermediate convolutional layers in CNNs trained on raw speech in an unsupervised manner. We argue that averaging across feature map values after ReLU activations yields interpretable time series data that summarizes encodings of phonetic features at each convolutional layer. This allows us to use standard acoustic phonetic measurements to test what properties of speech are encoded at what layer.

Acoustic analyses suggest that many acoustic properties are encoded in the fourth convolutional layer (Conv4). This layer features a clear period of frication noise (aperiodic vibration), a period of silence (corresponding to closure in stops) and a period of periodic vibration with some formant structure. Duration of the vocalic period is faithfully encoded in Conv4: periodic vibration between the two layers (Conv4 and final output) align almost perfectly. Visualizations in Figure 4 suggest that timing of other major acoustic properties (frication noise and silence) is also highly aligned between Conv4 and final output. Acoustic analysis of the fourth convolutional layer also suggests that F0 and intensity values (or acoustic envelope) are faithfully encoded in this layer.

Differences in the acoustic analysis of the two models — the bare WaveGAN and ciwGAN — suggest that the degree to which individual acoustic properties are encoded at various intermediate layers can differ somewhat across the models. The two models probed here differ in the number of training steps (12,255 in WaveGAN vs. 15,920 in ciwGAN), training data points (5,463 vs. 996), and consequently in the number of epochs (716 vs. 5,114). The structure of the Generator

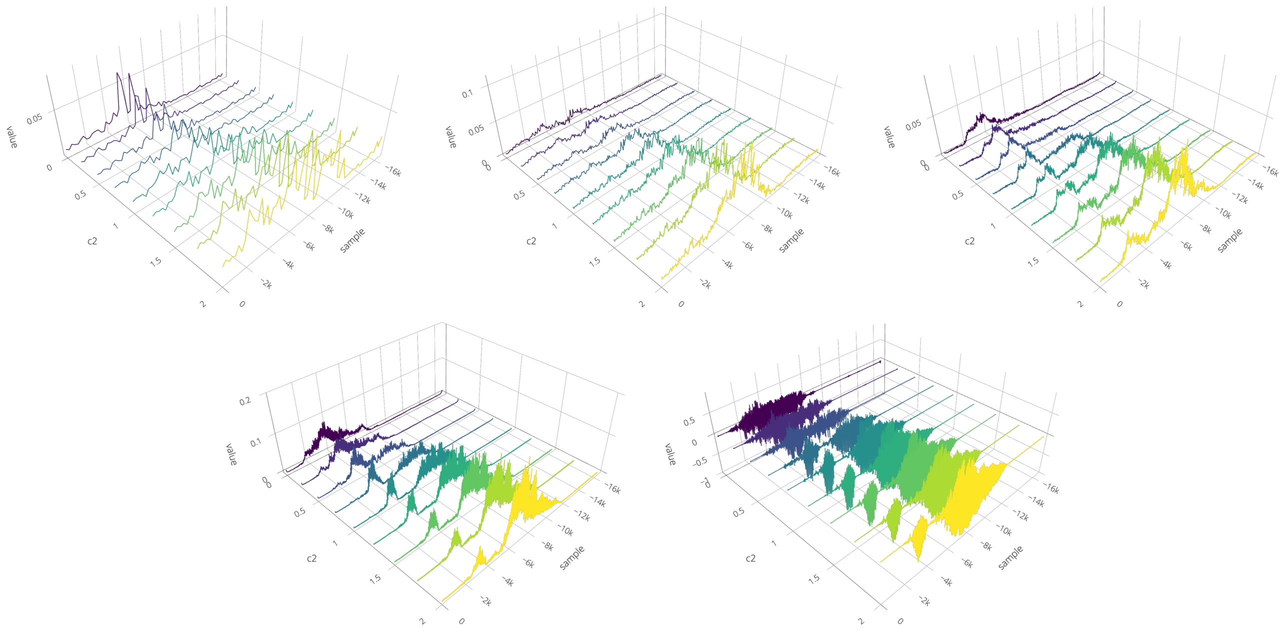

Fig. 14. Averaged values across feature maps after ReLU activation in the first (top left), second (top middle), third (top right), fourth (bottom left) convolutional layer, and the final waveform output (bottom right). At the values [0, 0] the final output layer can be transcribed as [ˈdɑji]. At the values of the latent code [0.625, 0], the output can be transcribed as [dəˈdaj]; at the value [1, 0] [təˈtʰɑjə]. For each convolutional layer, the graph represents 9 averaged values after ReLU activation where c_2 is linearly interpolated from 0 to 2 (in increments of 0.25) while c_1 is set to 0 and all other 98 latent z variables are held constant and limited to the training interval (-1,1) with uniform distribution. All outputs except in the final layer are upsampled with linear interpolation to total 16,384 samples (y-axis) to match the audio waveform output. Representation of the third, fourth, and final layer were cut off at 6100th sample because higher samples featured mostly silence. The figures illustrate how interpolating c_2 from 0 to 2 results in appearance of reduplication and how reduplication is encoded across the layers.

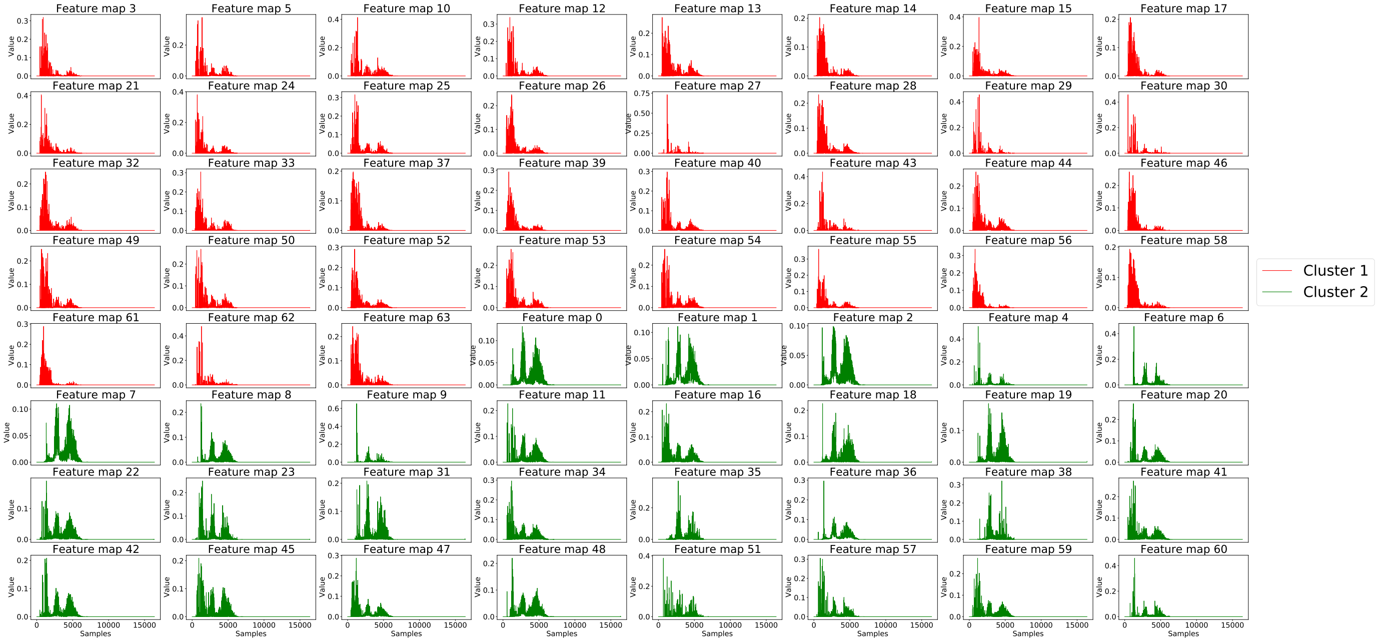

Fig. 15. Individual feature maps averaged over 500 instances of #sTV and 500 instances of #TV, clustered using spectral clustering. All feature maps exhibit an initial spike corresponding to the presence of a [s]-frication. However, the first cluster (red) encodes very little after the initial spike, while the second cluster (green) has significant activity corresponding to the rest of the sequence.

is identical across the models, except that in the ciwGAN architecture, the generator takes the latent code c in addition to the latent variables z as its input. The ciwGAN model trained on a computationally more complex process with substantially more epochs appears to encode formant structure in the fourth convolutional layer (Conv4) more faithfully than the bare WaveGAN model trained on #sTV. While the relationship between the formant structure in Conv4 and the actual output is complex, the fourth convolutional layer does feature a clear formant structure which is at least partly correlated with the

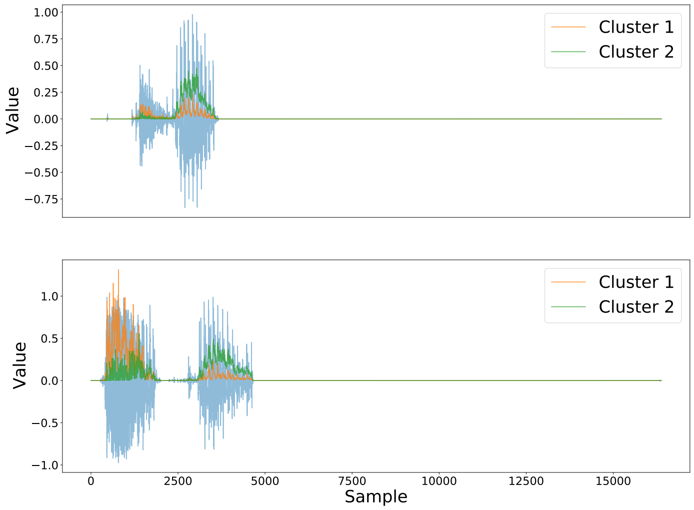

Fig. 16. The same clusters from Figure 15 averaged for a particular example of #TV (top, $z_{11} = 5$) and #sTV (bottom, $z_{11} = -15$), and plotted against the final output. Cluster 1 is much less activated than Cluster 2 in the #TV output, but becomes highly activated in exactly the region corresponding to [s] in the #sTV output.

final output (in F2 values).

The third convolutional layer is substantially more limited in what it can encode: with 1024 data points, its Nyquist frequency is 512 Hz. Formant structure is expectedly limited, but F0 and especially intensity data is faithfully encoded. The two tested models (WaveGAN and ciwGAN) also differ substantially in what is encoded in the third convolutional layer. Visualizations in Figure 14 suggest that intensity (acoustic envelope) is attested well into the second and even first convolutional layer. Vocalic periods and periods of reduced amplitude during closure are faithfully encoded in at least third and fourth convolutional layers in the ciwGAN model trained on reduplication. This stands in contrast to the bare WaveGAN, where the periods of vocalic vibration are not clearly distinguishable in the third convolutional layer (Figure 6).

Combining the proposed interpretation technique with manipulation and interpolation of individual latent variables with linguistically meaningful representations illustrates how individual variables in the latent space affect the activations at individual convolutional layers. Interpolating individual latent variables allows us to identify which activations in intermediate convolutional layers increase most substantially, thus identifying a causal relationship between the latent variables and activations in intermediate layers.

Averaged feature map values after ReLU activations summarize encodings at each convolutional layer and allow for standard acoustic analyses of the intermediate outputs. We can also probe individual feature maps by manipulating and interpolating individual latent variables. The effects of interpolation on individual feature maps is similar to its effect on the averaged values (Section IV-D).

Interpretation of intermediate stages by manipulating the latent space also suggests that different acoustic features (such as aperiodic frication noise or periodic vocalic vibration) can be encoded in separate feature maps. Clustering in Section IV-D1 suggests that some feature maps activate the frication part more strongly when the latent variable corresponding to [s] is manupulated to marginal levels, while in others the vocalic period is activated more strongly.

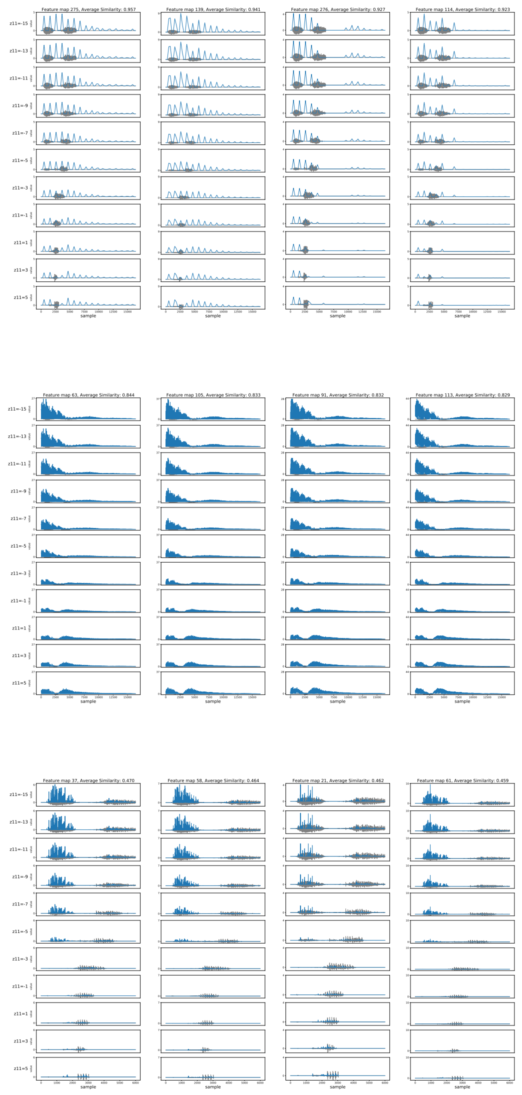

Fig. 17. Sets of individual feature maps after ReLU with minimal changes as determined by cosine distance from the values when $z_{11} = -15$. The feature maps are plotted at three convolutional layers: Conv1 (top), Conv3 (middle), Conv4 (bottom). Values of z_{11} are interpolated from -15 to 5 in increments of 1 for each convolutional layer and featue map (while other 99 z variables are kept constant).

The proposed technique allows several applications. The interpetation and visualization technique can serve as a diagnostic for improving the performance of CNNs trained on speech. The interpretation suggest that several acoustic properties relevant to speech perception (especially the formant structure of vowels) is encoded only in the final layer, primarily because the Nyquist frequency does not allow properties with higher frequencies to be encoded earlier in the structure of the Generator network. This suggests that introducing more layers capable of encoding properties with higher frequencies might improve performance of the model. Testing this hypothesis is left for future work.

The proposed technique can also serve for direct (albeit superficial) comparisons between intermediate convolutional layers and neural activity in the brain. A few parallels are immediately available: the output at the fourth convolutional layer (Conv4) resembles the complex auditory brain stem response when subjects are presented with acoustic vocalic stimuli (as in [42]). Also, parallel to the intensity values (or acoustic envolope) which are encoded high in the structure of the convolutional network (up to the second and even first convolutional layer in the ciwGAN), the acoustic envelope is encoded relatively high in the brain as well (in the auditory cortex) [43]. Detailed tests of parallels are left for future work, but the advantage of the proposed technique is that it outputs time-series data and enables testing of which acoustic properties are encoded at which layers. This information can be used for comparison between the convolutional networks and various neuroimaging techniques (which also output time-series data). The outputs of the proposed visualization technique should enable more informative and interpretable comparisons between the convolutional networks and the brains than only extracting the networks' activations and correlating them with outputs of brain imaging.

References

[1] H. van der Hulst, "Discoverers of the phoneme," in *The Oxford Handbook of the History of Linguistics*, K. Allan, Ed. Oxford University Press, 07 2013, pp. 167–191.

[2] M. K. C. MacMahon, "Orthography and the early history of phonetics," in *The Oxford Handbook of the History of Linguistics*, K. Allan, Ed. Oxford University Press, 07 2013, pp. 105–122. [Online]. Available: https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199585847.001.0001/oxfordhb-9780199585847-e-6

[3] M. Gasser, "Learning words in time: Towards a modular connectionist account of the acquisition of receptive morphology," Indiana University Bloomington, Tech. Rep., 1993. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.6474

[4] G. F. Marcus, S. Vijayan, S. Bandi Rao, and P. M. Vishton, "Rule learning by seven-month-old infants," *Science*, vol. 283, no. 5398, pp. 77–80, 1999. [Online]. Available: https://science.sciencemag.org/content/283/5398/77

[5] G. Beguš, "Generative adversarial phonology: Modeling unsupervised phonetic and phonological learning with neural networks," *Frontiers in Artificial Intelligence*, vol. 3, p. 44, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2020.00044/abstract

[6] G. Beguš, "Local and non-local dependency learning and emergence of rule-like representations in speech data by deep convolutional generative adversarial networks," 2020.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[8] J. Huang, J. Li, and Y. Gong, "An analysis of convolutional neural networks for speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4989–4993.

[9] A. Krug and S. Stober, "Visualizing deep neural networks for speech recognition with learned topographic filter maps," 2019.

[10] H. Muckenhirn, V. Abrol, M. Magimai-Doss, and S. Marcel, "Understanding and Visualizing Raw Waveform-Based CNNs," in *Proc. Interspeech 2019*, 2019, pp. 2345–2349. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2341

[11] A. Chowdhury and A. Ross, "Deepvox: Discovering features from raw audio for speaker recognition in degraded audio signals," 2020.

[12] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," 2020.

[13] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," 2018.

[14] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with sincnet," 2019.

[15] A. Krug and S. Stober, "Introspection for convolutional automatic speech recognition," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 187–199. [Online]. Available: https://www.aclweb.org/anthology/W18-5421

[16] A. Krug, R. Knaebel, and S. Stober, "Neuron activation profiles for interpreting convolutional speech recognition models," 2018.

[17] J. Millet and J.-R. King, "Inductive biases, pretraining and fine-tuning jointly account for brain responses to speech," 2021.

[18] D. Palaz, R. Collobert, and M. Magimai-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *CoRR*, vol. abs/1304.1018, 2013. [Online]. Available: http://arxiv.org/abs/1304.1018

[19] D. Palaz, M. Magimai.-Doss, and R. Collobert, "Analysis of cnn-based speech recognition system using raw speech as input," in *Proceedings of Interspeech*, ISCA. ISCA, 2015, pp. 11–15.

[20] P. Golik, Z. Tüske, R. Schlüter, and H. Ney, "Convolutional neural networks for acoustic modeling of raw time signal in lvcsr," in *Interspeech*, 2015, pp. 26–30,.

[21] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition," *Speech Communication*, vol. 108, pp. 15–32, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639316301625

[22] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-end convolutional neural network-based voice presentation attack detection," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, 2017, pp. 335–341.

[23] H. Muckenhirn, M. Magimai.-Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using cnns," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4884–4888.

[24] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR (workshop track)*, 2015. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a

[25] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[26] B. v. Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/interspeech.2020-1693

[27] Y.-A. Chung, H. Tang, and J. Glass, "Vector-Quantized Autoregressive Predictive Coding," in *Proc. Interspeech 2020*, 2020, pp. 3760–3764. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1228

[28] M. Chen and T. Hain, "Unsupervised Acoustic Unit Representation Learning for Voice Conversion Using WaveNet Auto-Encoders," in *Proc. Interspeech 2020*, 2020, pp. 4866–4870. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1785

[29] R. Eloff, A. Nortje, B. van Niekerk, A. Govender, L. Nortje, A. Pretorius, E. Biljon, E. van der Westhuizen, L. Staden, and H. Kamper, "Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks," in *Proc. Interspeech 2019*, 09 2019, pp. 1103–1107.

[30] C. Shain and M. Elsner, "Measuring the perceptual availability of phonological features during language acquisition using unsupervised binary stochastic autoencoders," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 69–85. [Online]. Available: https://www.aclweb.org/anthology/N19-1007

[31] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rylwJxrYDS

[32] C. Donahue, J. J. McAuley, and M. S. Puckette, "Adversarial audio synthesis," in *7th International Conference on Learning*

Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=ByMVTsR5KQ

[33] G. Beguš, "Ciwgan and fiwgan: Encoding information in acoustic data to model lexical learning with Generative Adversarial Networks," *Neural Networks*, vol. 139, pp. 305–325, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608021001052

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

[35] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 214–223. [Online]. Available: http://proceedings.mlr.press/v70/arjovsky17a.html

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777. [Online]. Available: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[37] J. S. Garofolo, L. Lamel, W. M Fisher, J. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 11 1993.

[38] G. Beguš, "Identity-based patterns in deep convolutional networks: Generative adversarial phonology and reduplication," 2020. [Online]. Available: https://arxiv.org/abs/2009.06110

[39] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]. version 5.4.06." Retrieved 21 February 2015 from http://www.praat.org/, 2015.

[40] Y. Xu, "Prosodypro — a tool for large-scale systematic prosody analysis," in *Proceedings of Tools and Resources for the Analysis of Speech Prosody (TRASP 2013), Aix-en-Provence, France*, 2013, pp. 7–10.

[41] Y. Xu and H. Gao, "Formantpro as a tool for speech analysis and segmentation / formantpro como uma ferramenta para a análise e segmentação da fala," *REVISTA DE ESTUDOS DA LINGUAGEM*, vol. 26, no. 4, pp. 1435–1454, 2018. [Online]. Available: http://www.periodicos.letras.ufmg.br/index.php/relin/article/view/13015

[42] T. C. Zhao and P. K. Kuhl, "Linguistic effect on speech perception observed at the brainstem," *Proceedings of the National Academy of Sciences*, vol. 115, no. 35, pp. 8716–8721, 2018. [Online]. Available: https://www.pnas.org/content/115/35/8716

[43] Y. Oganian and E. F. Chang, "A speech envelope landmark for syllable encoding in human superior temporal gyrus," *Science Advances*, vol. 5, no. 11, 2019. [Online]. Available: https://advances.sciencemag.org/content/5/11/eaay6279