MODELING DNN AS HUMAN LEARNER

By

Junrui Ni

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisor: Prof. Hasegawa-Johnson

December 2019

Abstract

In previous experiments, human listeners demonstrated that they had the ability to adapt to unheard, ambiguous phonemes after some initial, relatively short exposures. At the same time, previous work in the speech community has shown that pre-trained deep neural network-based (DNN) ASR systems, like humans, also have the ability to adapt to unseen, ambiguous phonemes after retuning their parameters on a relatively small set. In the first part of this thesis, the time-course of phoneme category adaptation in a DNN is investigated in more detail. By retuning the DNNs with more and more tokens with ambiguous sounds and comparing classification accuracy of the ambiguous phonemes in a held-out test across the time-course, we found out that DNNs, like human listeners, also demonstrated fast adaptation: the accuracy curves were step-like in almost all cases, showing very little adaptation after seeing only one (out of ten) training bins.

However, unlike our experimental setup mentioned above, in a typical lexically-guided perceptual learning experiment, listeners are trained with individual words instead of individual phones, and thus to truly model such a scenario, we would require a model that could take the context of a whole utterance into account. Traditional speech recognition systems accomplish this through the use of hidden Markov models (HMM) and WFST decoding. In recent years, bidirectional long short-term memory (Bi-LSTM) trained under connectionist temporal classification (CTC) criterion has also attracted much attention. In the second part of this thesis, previous experiments on ambiguous phoneme recognition were carried out again on a new Bi-LSTM model, and phonetic transcriptions of words ending with ambiguous phonemes were used as training targets, instead of individual sounds that consisted of a single phoneme. We found out that despite the vastly different architecture, the new model showed highly similar behavior in terms of classification rate over the time course of incremental retuning. This indicated that ambiguous phonemes in a continuous context could also be quickly adapted by neural network-based models.

In the last part of this thesis, our pre-trained Dutch Bi-LSTM from the previous part was treated as a Dutch second language learner and was asked to transcribe English utterances in a self-adaptation scheme. In other words, we used the Dutch model to generate phonetic transcriptions directly and retune the model on the transcriptions it generated, although ground truth transcriptions were used to choose a subset of all self-labeled transcriptions. Self-adaptation is of interest as a model of human second language learning, but also has great practical engineering value, e.g., it could be used to adapt speech recognition to a low-resource language. We investigated two ways to improve the adaptation scheme, with the first being multi-task learning with articulatory feature detection during training the model on Dutch and self-labeled adaptation, and the second being first letting the model adapt to isolated short words before feeding it with longer utterances.

Subject Keywords: Phoneme Category Adaptation, Human Perceptual Learning, Deep Neural Networks, Time-course, Long Short-Term Memory, Connectionist Temporal Classification, Second Language Learning, Articulatory Feature Detection, Multi-Task Learning, Semi-Supervised Learning

Acknowledgments

This thesis was co-advised by Prof. Scharenborg at the Delft University of Technology. The author would like to thank her for providing the retuning set and for her help with getting the forced alignment. The author would also like to thank her for all the insights and guidance she provided during all the experiments.

Contents

1. Introduction 1
1.1 The Time-Course of Phoneme Category Adaptation in Deep Neural Networks 1
1.2 Phoneme Category Adaptation Using Bi-LSTM and CTC 2
1.3 Second Language Learner Adaptation 3
2. Literature Review 5
2.1 Human Perceptual Learning 5
2.2 Second Language Learner 8
2.2.1 Perceptual Assimilation Model 8
2.2.2 Speech Learning Model 9
2.2.3 Native Language Magnet 11
2.3 Neural Network in Speech Recognition 12
2.3.1 Neural Networks and Back-Propagation 12
2.3.2 Convolutional Neural Networks 14
2.3.3 Recurrent Neural Networks, GRU and LSTM 15
2.3.4 Deep Neural Network in Speech Recognition 17
2.4 Visualization 20
2.4.1 Principal Component Analysis 21
2.4.2 Non-negative Factor Analysis 23
3. Methodology 26
3.1 Investigating the Time Course of DNN Perceptual Learning 26
3.2 Investigating the Time Course of Bi-LSTM Perceptual Learning 29
3.3 Modeling the Dutch ASR Model as a Second Language Learner 30
4. Experimental Results 34
4.1 Classification Rates For DNN Perceptual Learning 34
4.2 Inter-category Distance Ratio For DNN Perceptual Learning 36
4.3 Investigating Step-like Behavior For DNN Perceptual Learning 38
4.4 Visualizing Phoneme Boundary Shift For DNN Perceptual Learning 40

4.5 Recognition Rate for Bi-LSTM Perceptual Learning.....50
4.6 Visualizing Phoneme Boundary Shift For Bi-LSTM Perceptual Learning.....52
4.7 Modeling ASR as a Second Language Learner.....62
5. Discussion and Future Work.....67
6. Conclusion.....70

1. Introduction

1.1 The Time-Course of Phoneme Category Adaptation in Deep Neural Networks

When encountering a new speaker, both humans and speech recognition systems face the challenge of adapting to the pronunciation of that speaker and must do so in a way such that the new sounds are included into pre-existing sound categories. This process is defined as perceptual learning, and here, we focused on how pre-trained deep neural networks deal with perceptual learning of an ambiguous phoneme.

In a typical human lexically-guided perceptual learning experiment, listeners are first exposed to deviant phonemic segments in lexical contexts that constrain their interpretation, after which listeners have to decide on the phoneme categories of several ambiguous sounds on a continuum between two phoneme categories (e.g., [27, 95, 102, 109, 112, 114]). This way the influence of exposure to the deviant sound can be investigated on the phoneme categories in the human brain. In this paradigm [112], two groups of listeners are tested. One group of Dutch listeners was exposed to an ambiguous [l/ɹ] sound in [l]-final words such as appel (Eng: apple; appel is an existing Dutch word, apper is not). Another group of Dutch listeners was exposed to the exact same ambiguous [l/ɹ] sound, but in [ɹ]-final words, e.g., wekker (Eng: alarm clock; wekker is a Dutch word, wekkel is not). After exposure to words containing the [l/ɹ], both groups of listeners were tested on multiple steps from the same continuum of [l/ɹ] ambiguous sounds from more [l]-like sounds to more [ɹ]-like sounds. For each of these steps, they had to indicate whether the heard sound was an [l] or an [ɹ]. Percentage [ɹ] responses for the continuum of ambiguous sounds were measured and compared for the two groups of listeners. Lexically-guided perceptual learning shows itself as significantly more [ɹ] responses for the listeners who were exposed to the ambiguous sound in [ɹ]-final words compared to those who were exposed to the ambiguous sound in [l]-final words. A difference between the groups is interpreted to mean that listeners have retuned their phoneme category boundaries to include the deviant sound into their pre-existing phone category of [ɹ] or [l], respectively.

In previous work, it was shown that deep neural networks (DNNs) can also adapt to ambiguous speech by training on only a few examples of an ambiguous sound, with comparable behavior to humans in a similar setting [113]. However, the minimum amount of instances required for a DNN to adapt to this ambiguous sound remained unknown. Also, it would be useful to compare the time-course between humans and machines during this type of adaptation setting, thus helping to connect human perceptual learning with machine perception. Lastly, while visualizing the weights of DNN still remains an open topic, it would be interesting to show how the weights evolve through time as more adaptation tokens are fed. Therefore, the goal of this part of the thesis is three-fold: (1) to investigate the time-course of phoneme category adaptation in a DNN in more detail; (2) to connect between how humans and machines deal with phoneme category

adaptation; (3) to visualize the process of DNN phoneme category adaption over the time-course (by visualizing weights of the neural network).

1.2 Phoneme Category Adaptation Using Bi-LSTM and CTC

The goal of this part is very much the same as the first part, i.e., to observe the time-course of adaption and to visualize the time-course via neural network visualization. The main difference is that we changed the model for a human listener from a multi-layer DNN which is only capable of taking the context information of a fixed number of frames, to an end-to-end Bi-LSTM model that given the utterance of a word, would output the phonetic transcription of the utterance as a whole. Using Bi-LSTM and CTC loss for investigating machine perceptual learning is a better choice for connecting with human perceptual learning, for the following reasons:

1) In a typical human lexically-guided perceptual learning experiment, listeners could almost always refer to lexical contexts to constrain their interpretation [27, 66, 67, 102]. However, as the DNN model proposed previously could only take up to a fixed number of context frames (around 10 frames before and 10 frames after the current frame in our settings), the lexical context is, arguably, completely lost. Even if the training sounds from the re-tuning set were first grouped into words and fed to the DNN word by word, there is no guarantee that a simple multilayer DNN would make use of the contextual information implied by the data order to retune its internal representation. While it would certainly be interesting and useful to investigate how DNN applies “perceptual learning” to individual phonemes, the experimental setup is still not as close to how humans actually perform such adaptation.
2) Bi-LSTM, combined with CTC training and decoding [1, 53, 54], goes directly from raw spectrogram input to a sequence of phonemes (hence the notion of end-to-end model). Due to the use of input/output/forget gate, memory cell [49, 59] and bidirectional structure [53], every output unit inside of a Bi-LSTM model could in theory capture all the useful information from a whole input sequence. Note that this would be more similar to human perceptual learning experiments, as humans also hear the whole words, as opposed to individual phonemes, during perceptual learning experiments, and most likely use information from the whole utterance (for example, what the previous phonemes are and so what would the word most likely be) to determine whether the ending ambiguous sound [l/r] should be interpreted as [l] or [r]. Another nice property of the end-to-end Bi-LSTM-CTC model was that the CTC training algorithm is alignment free—it does not require an alignment between input and output sequence, very much like how humans perform adaptation in perceptual learning settings (i.e., they are almost never given a forced alignment of words during the perceptual learning experiments) [27, 66, 67, 102].

1.3 Second Language Learner Adaptation

Human L2 learning has been relatively well studied by three dominant theories (perceptual assimilation model [10, 11, 12, 86], speech learning model [35, 43], and native language magnet model [73]). The details of the three theories are reviewed in the next section, but PAM generally answers the question of why certain non-native phonetic contrasts are better discriminated than others before learning takes places; SLM investigates why certain L2 phonetic segments are better learned than others over the process of learning; NLM explains the perceptual magnet during infant's perceptual development.

Therefore, it would be interesting to ask how a machine (in this case, a deep neural network-based speech recognizer) would try to "learn" a second language, and how its performance could be improved via specific techniques used in human second language teaching. Here the word "learning" is restricted to the learning of recognition tasks only, and the performance is measured by phone error rate of the transcription.

ASR systems trained on one specific language usually perform poorly if asked to transcribe a different language, even if those two languages are closely connected [57, 110, 111]. Some difficulties include: (1) Some of the phones in the L2 language are not present in the L1 language, so in order to create those additional units, additional softmax layers must be created [110, 111]; without proper initialization, the recognition rate would be close to chance-like. (2) Even for the shared phonetic units of the two languages, there is little to no guarantee that the same IPA symbol corresponds to the same equivalent class of acoustic features [57]. However, they are mapped to the same softmax unit in the output layer and go through the same set of hidden representation transformations. (3) Recording conditions, speaker variations within cross-lingual data further complicates the issue as neural network-based ASR usually "prefers" input signals that are somewhat similar to the corpus it is trained on.

In the last part of the thesis, we use the self-training paradigm to adapt a relatively well-trained Dutch ASR model for transcribing English utterance. The model used here is similar to the one used in the previous part, i.e. a Bi-LSTM model trained using the CTC criterion. The self-training paradigm incorporates a three-step workflow[110]: 1) in the initialization step, missing L2 (English) phones are added to the softmax layer and initialized using a linear combination of phones in the L1 language (Dutch), based on linguistic knowledge about those phones 2) The initialized model is then asked to transcribe English utterances directly, and phone error rates are calculated using ground truth English phonetic transcriptions 3) Selected percentages (based on error rates) of the self-labeled utterances are used in a subsequent adaptation step to update the model weights.

The subsequent adaptation is capable of creating a statistically "better" model than the initialized model (see a conservative discussion in [110]); however, the decrease in error rate is still too small to notice any difference during visualization - i.e., a clear adaptation course, in terms of phone recognition rate of individual phones, is too hard to observe from the result. Therefore, two methods to improve the adaptation are proposed:

(1) In PAM, Best mentioned that non-native speech perception is strongly affected by listeners' knowledge of native phonological equivalent classes, i.e., the articulators [8]. It would be natural to extend the idea to machine L2 learning, i.e., utilizing articulatory feature detection as an auxiliary task for phone transcription. When training the model on Dutch (L1 learning state), the articulatory feature detection units are trained jointly with the phone output layers to implicitly learn an equivalent mapping from phones to sets of articulatory features.

(2) In most early L2 learning classroom settings, students tend to first learn individual words before they move onto longer sets of words or sentences. Analogously, it would be interesting to see if our Bi-LSTM-CTC ASR model would benefit from first adapting isolated words segmented from the adaptation set in the first pass before adapting to connected words and longer sentences in the subsequent pass. Also, as the CTC model is summing up all the possible alignments during the training phase [52], using shorter training utterances (i.e., isolated words) could help the model detect phone boundaries better.

2. Literature Review

The first two parts of the thesis involve how deep neural networks perform the task of lexically-guided perceptual learning in the special case of phoneme category adaptation (after re-tuning on a small set of ambiguous sounds). Therefore, the first section of this literature review will cover human perceptual learning [27, 102, 109].

The third part of the thesis involves how deep neural networks trained on one language could be modeled as a second language learner adapting to a new language. Therefore, in the second section of this literature review, topics on second language learner behavior will be discussed [8, 35, 74].

All speech learning models built throughout the thesis are based on deep neural networks. Therefore, in the third section of this literature review, neural networks and optimization will first receive a general discussion, followed by applications of deep neural networks in speech recognition [1, 52, 53], with an emphasis on the algorithms [52] and models [1] used in this thesis.

2.1 Human Perceptual Learning

In *Perceptual learning for speech* [109], Samuel and Kraljic reviewed several lines of research under two themes of perceptual learning: with Theme I being the case where the listener's ability to identify unfamiliar speech stimuli (nonnative phonetic contrasts; accented speech/dialects; degrade speech) improved after experience, and Theme II being the case where the listeners were presented with phonetically ambiguous stimuli and measurement of perceptual learning is more of phonetic boundary shift than of improved comprehension ability.

Some of the research in Theme I included discovering improved ability to distinguish between /r/ and /l/ for both bilingual and monolingual native Japanese speakers, after high-variability training of /r/-/l/ contrast [84], and showed both generalizations in terms of new speakers/tokens [83, 84] (although test subjects were significantly more accurate on familiar talkers [82]) and relatively long-lasting (three to six months as found out by retest) perceptual learning effect on modification to phonetic perception [82]. Moreover, the effects of perceptual learning in the case also enhanced the ability of Japanese speakers to produce the distinction of /r/-/l/ [47] that also showed long-term effects [17]. Similar results apply to native English speakers learning Mandarin tones (which are considered suprasegmental contrasts), with test subjects showing significantly improved ability in identifying the four tones [121], as well as to Chinese speakers improving their distinction of English contrasts (using either two-alternative forced-choice procedure or same/different discrimination procedure with non-significant differences) [36], suggesting the development of perceptual learning in both cases.

Accented speech can be hard to understand perceptually. Like the experiments in nonnative phonetic contrasts, using a high-variability method to train American listeners on accented speech from native Chinese speakers gave better generalization than the low-variability method, and in the latter case, improved perception could only be observed if training and testing speakers were the same, although the generalization of the former case also failed when another accent was encountered [16]. Further research [22] showed that less extensive training on only more than a dozen sentences (lasting only about one minute) of accented speech was able to improve the perception (instead of developing general strategies to cope with difficulty) of listeners in terms of matching visual targets. In the case of idiolect speech, reviewed research showed that people who were good at talker recognition performed better on speech from familiar talkers, while people who were not good at talker recognition in the first place showed no such difference, indicating speaker-specific learning effects [96], while reviewed research on deaf speech showed that experienced deaf speech listeners were good at recognizing deaf speech under different contexts even if the speaker was newly-encountered [89].

In the case of degraded speech after compression, fast perceptual learning could also be observed after 5 to 10 training sentences, when sentences were compressed to less than half of their original length [28]. Even more so, subsequent reviewed research showed that knowledge of the language of training sentences did not matter as much and was able to show that perceptual learning happens at the phonological level (instead of higher levels such as lexical processes) [98]. Experiments with different methods to degrade speech, such as noise vocoding, showed that hearing a clear version of speech before a vocoded version improved the level of perceptual learning [25]. Also, if nonword vocoded speech stimuli were short enough to retain in phonological STM, the same amount of perceptual learning, in terms of efficacy, happened as with real word stimuli [58]. As with nonnative phonetic contrast, generalization was also found to be better if vocoded speech came from different speakers (but the results varied as to the amount of misalignment) [118]. Similar perceptual learning results were also obtained with synthesized speech, and sleep between training and testing was found to consolidate the effect of perceptual learning [32].

Theme II of Samuel and Kraljic's review focused on phonetic retuning. This part of their review is more relevant to establishing the time-course of ambiguous phoneme adaptation experiments carried out in the first and second part of this thesis. Listeners were presented with ambiguous stimuli with context information, and perceptual learning happens as a shift in phoneme categorization, as listeners started to align the ambiguous phonemes with the context information [109]. In experiments related to lexically induced learning, listeners exposed to ambiguous sounds in the middle of /f/ and /s/ were able to use lexical information, such as whether interpreting that sound as /f/ would result in a real word, to guide their recognition of this ambiguous fricative, and thus in a subsequent categorization test would give more /f/ responses than /s/ (and vice versa) [95]. Even more so, listeners were able to generalize to words outside of the training set, showing adjustments to prelexical representations of fricatives [90], with perceptual learning happening automatically upon just hearing those words with ambiguous sounds (i.e., without explicitly identifying the sound or the word) [91], and such

learning can be quite long-lived and persistent [31, 70]. Also, in the case of training listeners with new “words”, with some of them containing ambiguous fricatives, perceptual learning results were better generated as listeners formed better lexical information about those novel words, under the guide of novel picture association [76]. However, interestingly, the “critical phonemes” in these studies also played a role, as fricatives /s/ and /f/ did not generalize well to new speakers and tend to be speaker-specific [67], while stops such as /d/ and /t/ did generalize (to both new voices and another pair of stops, in particular,/b/-/p/ [66]) and tend to be speaker-general [66, 67]. Other interesting effects of the perceptual system include the fact that the pronunciation of a new speaker was not learned unless they were encountered at the beginning, and that speaker-external factors were also not learned [69]. Several other mentioned experiments in the review, such as studying the perception of non-ambiguous sounds (which proved that learning is based on dynamic adjustments of the representations of sounds rather than transformation of signal) [24], production (production system did not change after perceptual learning) [68], and vowel space remapping (which showed very targeted shifts rather than boundary relaxation, and did not incur a complete remap of vowel space) [86] all provided useful aspects on this subject.

One last aspect of the review by Samuel and Kraljic covered audio-visual perceptual learning. Results showed that listeners associated ambiguous sounds with the face they saw articulating that sound (recalibration) [6], and repeated exposure to un-ambiguous sounds with corresponding articulating faces (selective adaptation) reduced reports of ambiguous sounds as the repeated unambiguous ones [6]. Also, recalibration and selective adaptation showed different time courses, with a monotonically descending course for selective adaptation, and curvilinear course (a rapid build-up, followed by a plateau, followed by a gradual decline) for recalibration [120].

One important focus of this thesis is on the time-course of perceptual learning, which was studied in detail in the following two papers.

In the paper *The Time Course of Perceptual Learning* [102], the author fed one group with ambiguous [s/f] sounds in /s/-final words and another group with ambiguous [s/f] sounds in /f/-final words and utilized the visual-world eye-tracking paradigm, with displays that could be used as either training trail or test trial. Their set of stimuli contained the same number (20) of training items where the fricative could only be interpreted as either /s/ or /f/ (not both), temporary minimal pairs where the fricative could be interpreted as both but can be disambiguated using future context, and 20 minimal pairs which did not have any disambiguating context and thus were used for testing. The stimuli were presented in 20 mini-blocks, each consisting of one training item, one contrast item (e.g. natural /s/ for /f/-bias group), both members of a minimal pair (ambiguous fricative + natural contrast item) and one member of a temporal minimal pair. According to the experimental results from eye-tracking distance and fixation time, perceptual learning was found to occur at roughly mini-block 10, which includes 10 training items and 10 [s/f]-bearing temporary minimal pairs.

In the paper *Processing and Adaption to Ambiguous Sounds during the Course of Perceptual Learning* [27], the authors investigated the perception and processing of words with ambiguous [f/s] sound during the course of lexically-guided perceptual learning and tried to answer whether these ambiguous words were processed as natural stimuli, and what the time course was like. They created stimuli of prime-target pairs, where prime words are /f/ and /s/ final words of either natural sound or ambiguous sound (with a five-step continuum chosen using a pilot test), and target words being words that are semantically related words with neither /f/ or /s/ nor any ambiguity. They performed a lexical decision where listeners decide if the word is a real word with recorded response time, followed by a phonetic categorization task on the five-step /f/-/s/ continuum where listeners decide whether the word ends with /s/ or /f/. Some conclusions include the fact that prime words with ambiguous sounds had a lower acceptance rate than natural words and had a longer response time (with the relationship that primes with lower acceptance rate needing longer processing time) but without affecting the processing of the following target word, as well as that participants exposed to ambiguous /s/ sounds in /s/-final words gave more /s/ responses that ambiguous /f/ sounds in /f/-final words (showing lexically guided perceptual learning). Their most related conclusion to this thesis was that recognition of ambiguous words did become more natural-like towards the end of the exposure, with an increasing acceptance rate. Moreover, this happened after approximately 15 items, showing a step-like manner, which was on par with the results from related literature [66, 67, 102].

2.2 Second Language Learner

Second language learners often have difficulty perceiving the phonetic differences among contrasting consonants or vowels that are not distinct in their native language [51, 104, 119]. Three theoretical frameworks, which are Best's perceptual assimilation model (PAM), Flege's speech learning model (SLM), and Kuhl's Native Language Magnet model (NLM), offer explanations as to how and why one's native speech system affects the learning of sounds from a second language. PAM focuses on how non-native contrasts are mapped to native language perceptual space before formal second language learning takes place; SLM focuses on how second language acquisition evolves over time; NLM specifically targets the formation of language-specific pattern during the development of infant's perceptual space.

2.2.1 Perceptual Assimilation Model

Classic views such as critical early tuning [30] failed to explain why successful contrast tuning could be successful during adulthood [83, 84] and why adult discrimination of nonnative contrasts is not uniformly poor [7, 104]. The fundamental premise of PAM is that non-native segments tend to be perceived according to their similarities to, and discrepancies from, native segments that are closest in the native phonological space [9]. PAM further hypothesizes that non-native contrasts are best discriminated if perceived as phonologically equivalent to a native

contrast, still well discriminated if perceived as phonetic distinctions between good and poor examples of a single native consonant, and much worse if the two non-native contrasts are phonetically equivalent to a single native consonant [8, 9].

PAM is also directly related to articulatory phonology and claimed that non-native speech perception is strongly affected by listeners' knowledge of native phonological equivalent classes. Non-native phones can be assimilated to native phones based on articulators, constriction locations and/or constriction degrees used [9, 10, 11, 12].

According to [106], a single non-native phone can be assimilated into the native system in three ways: (1) categorized exemplar of native phoneme, (2) uncategorized phoneme that falls between native phonemes, (3) non-assimilable sound with little similarity to any native phoneme. According to this, several pairwise assimilations of two non-native phones exist: (1) two non-native phones assimilated to two different native phonemes (two category assimilation), (2) two non-native phones assimilated to a single native phoneme (single category assimilation), (3) two non-native phones assimilated to a single native phoneme with different levels of fit (category goodness difference), (4) uncategorized-categorized pair, (5) two uncategorized segments, (6) two non-assimilable sounds.

Therefore, PAM predicts that NA sounds, as unaffected by native phonology, can have good discrimination if the sounds themselves are different enough. TC and UC should also be discriminated well. CG would be well enough if one is a good fit and one is a bad fit but otherwise hindered, and SC would most likely be hindered as well, with the famous /r/-/l/ case for Japanese speakers. UU is affected by non-native contrast similarity and nearby native phones and can range from fair to good [8, 9].

2.2.2 Speech Learning Model

Speech Learning Model tackles the question as to why individuals learn or fail to learn to accurately perceive and produce phonetic segments in the second language [35]. The research by Flege focused on determining if there are "unlearnable" L2 sounds, and if so, are those sounds limited to adults, as well as how the perception of speech sounds encountered on the phonetic surface of an L2 influence their eventual production.

Some prior research and hypotheses that constitute the development of SLM prior to its formal establishment include, but are not limited to, several hypotheses:

(1) Critical Period Hypothesis (CPH), which, apart from the proposed notion for a critical period for primary language acquisition, also casually included the following claim: past a critical age, it is difficult to learn L2 without a foreign accent (FA) [79].
(2) Contrastive Analysis Hypothesis (CAH), which claimed that the more different the two languages (L1 and L2) get, the greater the difficulty of learning will be [122].

(3) Categorical Perception (CP), which claims for an absence of clearly perceived changes within a category as the stimuli cross a category boundary [107].

However, Flege pointed out through experiments that most of the above early theories regarding L2 suffered severe flaws. Some examples include: for the critical period hypothesis, a group of immigrants ([34] studied Italians; [46] studied Koreans) with different ages of arrival in North America was tested on their foreign accent, and the results showed that while few subjects with entry age after 12 were without FA, less than half who entered prior to 12 still demonstrated foreign accent [84]. Also, studies on Korean children in North America showed that FA was still detectable after 3-5 years of emergence [37, 84], and thus FA was definitely not the result of passing a critical period. Also, CPH cannot explain why some adult L2 learners manage to speak without FA [15]; for CAH, experiments on different groups of Americans with French experience [45] showed that adult L2 learners can be more successful at producing a "new" vowel sound that was very different from any other sound in the L1 inventory while having difficulty with a similar vowel, which was exactly the opposite of what CAH suggests; for CP, Flege found out that native English monolinguals displayed ability to detect within-category variations of French-accented /tu/. Other research on the commutability of existing abstract phonemic features [40] and learning of new phonemic features [41, 88, 94] further complicate L2 acquisition, as while Arabic learners of English did not recombine existing abstract features to produce a new L2 sound [40], it can be difficult for an L1 speaker to acquire a new, abstract feature in L2 [88], but the difficulty was related to age of learning [41, 94]. Furthermore, learning an L2 was also found to affect the production of L1, and the effects seemed to be stronger for early learners [124].

The SLM was therefore developed to make sense of empirical results that contradicted some earlier theories. Some basic premises of SLM include [35, 43]:

(1) L2 learners can, in time, perceive L2 phonetic properties.
(2) L2 speech learning takes time and is influenced by nature of input (as in L1).
(3) L2 production is guided by perceptual representations stored in long-term memory.

Further propositions and hypotheses include:

(4) The process and mechanism that guide L1 speech acquisition (such as the ability to form new phonetic categories) remain intact throughout life [44, 45].
(5) The L1 and L2 phonetic elements exist in common phonological space and mutually influence each other [45].
(6) The greater the perceived dissimilarity (as measured in perceptual experiments) of an L2 sound from the closest L1, the more likely a new category will be formed.
(7) Category formation for L2 sound becomes less likely through childhood as representations for neighboring L1 sound develop.
(8) When a category is not formed for L2 sound because it is too similar to an L1 sound, merged L1-L2 takes place.

The implication of (1), (2), (5) and (6) is that at the early stage of L2 learning, L2 vowels that are rated as perceptually similar (as L1 vowels) could be produced quite well, but not those that are

perceptually dissimilar, but over time, the production performance of those dissimilar ones would surpass those similar ones as dissimilar vowels tend to form new categories while those similar will not. While (8) is considered as an implication of (5), another implication is that when categories are created for an L2 vowel, the L2 vowel and its close L1 vowels will try to push apart in the perceptual space to minimize confusion, but that may make the production of one or both sounds less accurate [38].

2.2.3 Native Language Magnet

Infant speech perception and speech production changed from language-general to language-specific (with respect to the ambient language) after about a year [73, 123]. NLM suggests that linguistic experience alters the perceptual space of speech stimuli, in terms of “magnet effect”, where the most representative instances of a phonetic category function like magnets, attracting nearby members of the same category [72, 73], and therefore makes it difficult to discriminate between the “prototype” and those other sounds [74]. By comparing the adult perception of prototypes and non-prototypes of native and foreign vowels, with those of infants, studies showed that this magnet effect formed as early as 6 months into the infant’s life [71, 72, 103] (and 10 -12 months for consonants [103]). Other studies of the perceptual space using MDS and a synthesized set of syllables at equal distances also showed that perceptual space is distorted and shrunk around the best “representations” while stretched near the category boundary [62].

The formal theory of NLM incorporates three phases of speech development [73, 74]:
(1) Phase one refers to the infants’ born abilities to partition sounds into gross categories, separated by natural boundaries, with no dependence on a specific language (i.e., just basic auditory perceptual processing mechanism).

(2) Phase two refers to the perception of a 6-month-old infant. As infants have heard quite an amount of ambient speech, they start to develop different representations of the properties of vowels in their memory, and as their ambient language differs, their representation of the vowel system also differs, and start to show language-specific magnet effects.

(3) Phase three refers to the later stage when magnet effects caused certain acoustic differences to be minimized and others to be maximized, thus erasing some of the natural boundaries that existed in earlier phases, especially those contrasts that are not in their native or ambient language. At this phase, the warped perceptual space starts to take place.

NLM explains how speech perception changes in the infant stage [123] and why adults have certain perceptual behaviors with regard to sounds in a foreign language [10, 33] (for example, it explains Japanese speakers /r/-/l/ difficulty by predicting that their Japanese category prototype will attract both /r/ and /l/ [74]).

2.3 Neural Network in Speech Recognition

2.3.1 Neural Networks and Back-Propagation

Deep neural networks are powerful learning machines made up of rather simple building blocks such as matrix-vector multiplication and scalar nonlinearities [29]. For simplicity, this section will only derive some of the formulas for learning a two-layer network; the algorithm for learning a deep neural network with more than two layers should be rather similar.

Suppose the input vector is $\mathbf{x}$ of dimension d, and denote $\mathbf{x}^{'}$ of dimension $d+1$ as $\mathbf{x}$ concatenated with an extra one, used for adding a bias to the output vector $\mathbf{a}$. Then the initial output after going through the first layer of the neural net can be written as:

$$\mathbf{a} = \mathbf{U}\mathbf{x}^{'} \quad (1)$$

where $\mathbf{U}$ is a linear transformation matrix of $r \times (d+1)$, and r is the dimension of $\mathbf{a}$. Note that the extra input dimension effectively adds a bias to the output. The final output $\mathbf{y}$ from this first hidden layer is obtained after applying an element−wise scalar nonlinear function to $\mathbf{a}$, i.e.,

$$\mathbf{y} = f(\mathbf{a}) \quad (2)$$

Some popular non-linear functions for the hidden layer include:

a) Sigmoid, denoted as $1/(1+e^{-x})$
b) Tanh, denoted as $(e^{x}-e^{-x})/(e^{x}+e^{-x})$
c) ReLU, denoted as $max\{0,x\}$

The second layer of the network takes the output $\mathbf{y}$ from the previous layer, concatenates with a scalar one again to add bias, and multiplies $\mathbf{y}^{'}$ by a second weight matrix $\mathbf{V}$ of $(r+1) \times q$ to get the output vector $\mathbf{b}$, i.e.

$$\mathbf{b} = \mathbf{V}\mathbf{y}^{'} \quad (3)$$

The final output from this second layer is denoted as

$$\mathbf{z} = g(\mathbf{b}) \quad (4)$$

where g denotes the output (nonlinear) function. Some common output functions include sigmoid, which is used for binary classification and softmax:

$$\mathbf{z}_l = e^{\mathbf{b}_l} / \sum_m e^{\mathbf{b}_m} \quad (5)$$

which is used for multi-class classification.

In order to train a two-layer neural network, we need to optimize its parameters by minimizing some error metrics. For linear outputs, the error is chosen to be the mean squared error between the target ζ_i and the output $\mathbf{z}_i$, i.e.

$$L = \frac{1}{2N}\sum_{i=1}^{N} |\zeta_i - \mathbf{z}_i|^2 \quad (6)$$

For softmax output and one-hot target vectors, the cross-entropy loss is used, i.e.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{l}\zeta_{l,i}\log z_{l.i} \quad (7)$$

where $\zeta_{l,i}$ are entries of the one-hot target vector ζ_i, and $z_{l,i}$ are the probability outputs for each class from the softmax output layer of the network.

With the loss defined, training the weight matrix $\mathbf{U}$ and $\mathbf{V}$ is achieved via gradient descent. Gradient descent is an iterative update algorithm that uses the derivative of the loss function to update the parameters of the model. Suppose $u_{k,j}$ and $v_{l,k}$ are elements in the two weight matrices in the previous iteration, and $\hat{u}_{k,j}$ and $\hat{v}_{l,k}$ are the weights in the next iteration, then the update formula is:

$$\hat{u}_{k,j} = u_{k,j} - \eta\frac{\partial L}{\partial u_{k,j}} \quad (8)$$
$$\hat{v}_{l,k} = v_{l,k} - \eta\frac{\partial L}{\partial v_{l,k}}$$

where η is called the learning rate, which is usually tweaked for best convergence.

To actually obtain the gradient of the loss with respect to $u_{k,j}$ and $v_{l,k}$, backpropagation is used, which can be considered as continuously applying chain rule until it reaches the partial derivative of interest. Therefore, we have

$$\frac{\partial L}{\partial v_{l,k}} = \frac{1}{N}\sum_{i=1}^{N}(\frac{\partial L}{\partial b_{l,i}})(\frac{\partial b_{l,i}}{\partial v_{l,k}}) = \frac{1}{N}\sum_{i=1}^{N}\epsilon_{l,i}y_{k,i} \quad (9)$$

where the second term in the summation comes from the fact that

$$b_{l,i} = \sum v_{l,k}y_{k,i} \quad (10)$$

and thus

$$\frac{\partial b_{l,i}}{\partial v_{l,k}} = y_{k,i} \quad (11)$$

The first term is denoted as

$$\epsilon_{l,i} = \frac{\partial L_i}{\partial b_{l,i}} \quad (12)$$

which equals to

$$z_{l,i} - \zeta_{l,i} \quad (13)$$

if cross-entropy loss is used, and equals to

$$(z_{l,i} - \zeta_{l,i})g^{'}(b_{l,i}) \quad (14)$$

if mean squared error with nonlinear function g is used. To calculate the gradient with respect to the weight matrix $u_{k,j}$ of the first layer, the loss is further back-propagated to the first layer, and so we have:

$$\frac{\partial L}{\partial u_{k,j}} = \frac{1}{N}\sum_{i=1}^{N}(\frac{\partial L}{\partial a_{k,i}})(\frac{\partial a_{k,i}}{\partial u_{k,j}}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{k,i}x_{j,i} \quad (15)$$

where

$$\delta_{k,i} = \frac{\partial L_i}{\partial a_{k,i}} = \sum_{l=1}^{q}\epsilon_{l,i}v_{l,k}f^{'}(a_{k,i}) \quad (16)$$

2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs), which were first proposed in [2] to recognize spatio-temporal bipolar patterns associatively, are now widely applied in computer vision for image classification [117], object detection [106], super-resolution [77], etc. They are also used in speech recognition models for learning temporal and frequency features from spectrograms before feeding into recurrent layers [1]. The filters defined by the convolutional layers try to learn a set of translation-invariant features from the given input [65], thus avoiding hand-crafting various kinds of matched filters.

There are two additional kinds of layers in a CNN, the convolutional layer and pooling layer. Given input $x[n_1, n_2, j]$, where n_1 denotes row dimension out of N_1, n_2 denotes column dimension out of N_2, and j denotes channel dimension out of C, the convolutional layer is defined as a per-channel 2D convolution between $x[n_1, n_2, j]$ and convolution filter set $u[n_1, n_2, , j, k]$ of size $M_1 \times M_2 \times C \times C_{out}$, as:

$$\begin{aligned} a[n_1, n_2, k] &= u[n_1, n_2, j, k] * x[n_1, n_2, j] \\ &= \sum_j \sum_{m_1} \sum_{m_2} u[n_1 - m_1, n_2 - m_2, j, k] x[m_1, m_2, j] \end{aligned} \tag{17}$$

where $a[n_1, n_2, k]$ is the output feature map from the convolutional layer, with channel size C_{out}. The output usually goes through some activation function; here for simplicity, only the ReLU function is considered.

Another type of layer is the pooling layer. Here only max-pooling will be discussed; other types of pooling are similar enough. Given the previous post-ReLU activation output, max-pooling is defined as:

$$y_[n_1, n_2, k] = max_{(m_1, m_2) \in A(n_1, n_2)} max(0, a[m_1, m_2, k]) \tag{18}$$

with

$$A(n_1, n_2) = (m_1, m_2) : n_1 M \leq m_1 < (n_1 + 1)M, n_2 M \leq m_2 < (n_2 + 1)M \tag{19}$$

where M is the pooling stride. Note that there are no trainable weights associated with pooling layers.

To train the filters $u[m_1, m_2, j, k]$, backpropagation followed by gradient descent is used. Suppose the loss for a single training token is L_i, according to the chain rule, we have

$$\frac{\partial L_i}{\partial u[m_1, m_2, j, k]} = \sum_{n_1} \sum_{n_2} \left(\frac{\partial L_i}{\partial a_i[n_1, n_2, k]} \frac{\partial a_i[n_1, n_2, k]}{\partial u[m_1, m_2, j, k]} \right) \tag{20}$$

As

$$a_i[n_1, n_2, k] = \sum_j \sum_{m_1} \sum_{m_2} u[m_1, m_2, j, k] x[n_1 - m_1, n_2 - m_2, j] \quad (21)$$

It is easy to see that

$$\frac{\partial a_i[n_1,n_2,k]}{\partial u[m_1,m_2,j,k]} = x_i[n_1 - m_1, n_2 - m_2, j] \quad (22)$$

Therefore, it is easy to get

$$\begin{aligned} \frac{\partial L_i}{\partial u[m_1, m_2, j, k]} &= \sum_{n_1} \sum_{n_2} \delta_i[n_1, n_2, k] x_i[n_1 - m_1, n_2 - m_2, j] \\ &= \delta_i[m_1, m_2, k] * x_i[-m_1, -m_2, j] \end{aligned} \quad (23)$$

where

$$\delta_i[n_1, n_2, k] = \frac{\partial L_i}{\partial a_i[n_1, n_2, k]} \quad (24)$$

is the loss back-propagated to the pre-activation output $a_i[n_1, n_2, k]$. Note that this is a correlation with respect to back-propagated error and input feature map.

The derivative with respect to $x_[n_1, n_2, j]$ can be similarly computed as

$$\begin{aligned} \frac{\partial L_i}{\partial x[m_1, m_2, j]} &= \sum_k \sum_{n_1} \sum_{n_2} \left(\frac{\partial L_i}{\partial a_i[n_1, n_2, k]} \frac{\partial a_i[n_1, n_2, k]}{\partial x[m_1, m_2, j]} \right) \\ &= \sum_k \delta_i[m_1, m_2, k] * u[-m_1, -m_2, j, k] \end{aligned} \quad (25)$$

which is another correlation, but with respect to back-propagated error and filter weights.

Suppose that during backpropagation, the gradient has now been propagated to $y_i[o_1, o_2, k]$, then the previous term $\delta_i[n_1, n_2, k]$ could be calculated as

$$\delta_i[n_1, n_2, k] = \sum_{o_1} \sum_{o_2} \left(\frac{\partial L_i}{\partial y_i[o_1,o_2,k]} \right) \left(\frac{\partial y_i[o_1,o_2,k]}{\partial a_i[n_1,n_2,k]} \right) \quad (26)$$

where the last term could be simply calculated as 1 if $a_[n_1, n_2, k]$ "survives" the max-pooling and ReLU activation, and 0 otherwise.

2.3.3 Recurrent Neural Networks, GRU and LSTM

The recurrent neural network (RNN) and its variants are widely used in sequence learning tasks such as machine translation [19] and speech recognition [1, 19, 53]. Given an input sequence $\mathbf{x} = \{x_1, ..., x_T\}$, an RNN [53] computes the hidden representations $\mathbf{h} = \{h_1, ..., h_T\}$ and output vectors $\mathbf{y} = \{y_1, ..., y_T\}$ iteratively as:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (27)$$
$$y_t = W_{hy}h_t + b_y \quad (28)$$

where W_{xh} is the input weight matrix, W_{hh} is the hidden weight matrix, W_{hy} is the output weight matrix and b_h/h_y are bias vectors for hidden representation/output. The function H is the hidden activation function, which is usually a sigmoid function [53].

One problem with uni-directional RNNs is that for a time step t within the forward pass, that time step could only access information prior to itself, i.e., from 0 to $t-1$. For speech recognition, it would usually be helpful to gain access from future context as well [23]. Therefore, two separate layers are used in bi-directional RNN [116], one for processing forward sequence and one for processing backward sequence, as follows:

$$\vec{h}_t = H(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \quad (29)$$
$$\overleftarrow{h}_t = H(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (30)$$
$$y_t = H(W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y) \quad (31)$$

However, naive RNN structure often suffers from the exploding/vanishing gradient problems [5, 99] and thus possesses little capability of learning long-range contextual information.

Therefore, Long Short-Term Memory (LSTM) units are proposed [49, 59]. A common LSTM unit incorporates a memory cell, an input gate, an output gate and a forget gate [49]. The architecture permits LSTM to bridge between two input events with a large time lag, relatively independent of the intervening time steps [49]. The formula for calculating the gate values, the cell state, and the hidden representation of an LSTM network [49, 53] is defined as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (32)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (33)$$
$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (34)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (35)$$
$$h_t = o_t tanh(c_t) \quad (36)$$

where i, f, c, o are the input gate, forget gate, cell state and output gate, respectively.

Combining LSTM with bi-directional architecture gives Bi-LSTM [1, 53], which forms the backbone of speech recognition models for the second and third part of this thesis.

Another RNN unit worth mentioning is called Gated-Recurrent Unit (GRU) [20]. Like LSTM, GRU also uses gates to modulate information flow within the unit. It consists of two gates, an update gate and a reset gates, which jointly decide how much of the previous activation and candidate activation should be recorded as the current state. However, unlike LSTM, GRU does not have control of the amount of current state exposed to output [21]. The formula for the forward pass of GRU is defined as:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (37)$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (38)$$
$$h_t = (1 - z_t)h_{t-1} + z_t tanh(W_{xh}x_t + W_{hh}(r_t h_{t-1}) + b_h) \quad (39)$$

where z_t is the update gate vector and r_t is the reset gate vector.

2.3.4 Deep Neural Network in Speech Recognition

One important algorithm that allows recurrent neural networks (RNNs) to learn an end-to-end mapping from raw speech input space $\mathbf{X} = (\mathbb{R}^m)^*$ (usually spectrogram or Mel-spectrogram) to the label space $Z = L^*$ of phonetic transcription without any pre-segmentation or post-processing is called Connectionist Temporal Classification (CTC) [52, 54]. It assumes that the target sequence length U is at most as long as the input sequence length T, and learns a probabilistic distribution over all possible label sequences given the input sequence. The derivations below came from the original CTC paper [52].

The CTC label space consists of one extra label than the label originally in L, known as the blank label. This, together with the original $|L|$ labels, allows all possible label alignment with respect to the input sequence. Denote this new set of labels as L', then given a specific sequence of softmax output π from the RNN, we have

$$p(\pi|\mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t \quad (40)$$

where y_k^t is the probability of label k at time t, and the outputs are conditionally independent.

To map from π of L'^T to the actual transcription of $L^{\leq T}$, the many-to-one mapping B is used, which removes blank symbols from π, and squashes other repeating symbols that are not separated by a blank into one single symbol. Using B, the probability of a given labeling $\mathbf{l}$ of $L^{\leq T}$ is defined as:

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in B^{-1}(\mathbf{l})} p(\pi|\mathbf{x}) \quad (41)$$

which is the total probability of all paths corresponding to $\mathbf{l}$.

The objective function for training the CTC network is again based on maximum likelihood. The CTC Forward-Backward Algorithm offers an efficient way to calculate the probability $p(\mathbf{l}|\mathbf{x})$ as follows:

1. First, for a labeling $\mathbf{l}$, denote the forward variable $\alpha_t(s)$ as the total probability of $\mathbf{l}_{1:s}$ at time t:

   $$\alpha_t(s) = \sum_{\pi \in N^T; B(\pi_{1:t}) = \mathbf{l}_{1:s}} \prod_{t'=1}^{t} y_{\pi_t}^{t'} \quad (42)$$

   Using a modified label sequence $\mathbf{l}'$ with blanks added to beginning and end as well as between every two non-blank symbols, and allowing transition only between blank and non-blank labels or between two distinct non-blank labels, initializing and updating $\alpha_t(s)$ can be carried out as:

a. Initialization:

$$\alpha_1(1) = y_b^1 \quad (43)$$
$$\alpha_1(2) = y_{\mathbf{l}_1}^1 \quad (44)$$
$$\alpha_1(s) = 0, \forall s > 2 \quad (45)$$

b. Update:

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1))y_{\mathbf{l}'_s}^t$$
$$\text{if} \quad \mathbf{l}'_s = b \quad \text{or} \quad \mathbf{l}'_{s-2} = \mathbf{l}'_s; \quad (46)$$

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2))y_{\mathbf{l}'_s}^t$$
$$\text{otherwise} \quad (47)$$

$$\alpha_t(s) = 0 \qquad \forall s < |\mathbf{l}^s| - 2(T-t) - 1 \quad (48)$$

2. Similarly, the backward variable $\beta_t(s)$ can be defined as the total probability of $\mathbf{l}_{s:|\mathbf{l}|}$ at time t:

$$\beta_t(s) = \sum_{\pi \in N^T; B(\pi_{t:T}) = \mathbf{l}_{s:|\mathbf{l}|}} \prod_{t'=t}^{T} y_{\pi'_t}^{t'} \quad (49)$$

and again using the modified $\mathbf{l}'$, the initialization and update of $\beta_t(s)$ can be calculated as:

c. Initialization:

$$\beta_T(|\mathbf{l}'|) = y_b^T \quad (50)$$
$$\beta_T(|\mathbf{l}'| - 1) = y_{\mathbf{l}_{|\mathbf{l}|}}^T \quad (51)$$
$$\beta_T(s) = 0, \forall s < |\mathbf{l}'| - 1 \quad (52)$$

d. Update:

$$\beta_t(s) = (\beta_{t+1}(s) + \beta_{t+1}(s+1))y_{\mathbf{l}'_s}^t$$
$$\text{if} \quad \mathbf{l}'_s = b \quad \text{or} \quad \mathbf{l}'_{s+2} = \mathbf{l}'_s \quad (53)$$

$$\beta_t(s) = (\beta_{t+1}(s) + \beta_{t+1}(s+1) + \beta_{t+1}(s+2))y_{\mathbf{l}'_s}^t$$
$$\text{otherwise} \quad (54)$$

$$\beta_t(s) = 0 \qquad \forall s > 2t \quad \text{and} \quad \forall s > |\mathbf{l}'| \quad (55)$$

The probability $p(\mathbf{l}|\mathbf{x})$ is simply the sum of $\alpha_T(|\mathbf{l}'|)$ and $\alpha_T(|\mathbf{l}' - 1|)$.

Maximum likelihood training is carried out by first calculating the derivative with respect to network outputs $\mathbf{y}_k^t$. Using

$$\alpha_t(s)\beta_t(s) = \sum_{\pi \in B^{-1}(\mathbf{l}); \pi_t = \mathbf{l}'_s} y_{\mathbf{l}'_s}^t \prod_{t=1}^{T} y^t{}_{\pi_t} \quad (56)$$

we can get

$$\frac{\partial p(\mathbf{l}|\mathbf{x})}{\partial y_k^t} = \frac{1}{(y_k^t)^2} \sum_{s \in lab(\mathbf{l},k)} \alpha_t(s)\beta_t(s) \quad (57)$$

where

$$lab(\mathbf{l}, k) = \{s : \mathbf{l}'_s = k\} \quad (58)$$

and therefore

$$\frac{\partial \log(p(\mathbf{l}|\mathbf{x}))}{\partial y_k^t} = \frac{1}{p(\mathbf{l}|\mathbf{x})} \frac{\partial p(\mathbf{l}|\mathbf{x})}{\partial y_k^t} \quad (59)$$

The gradient with respect to u_k^t as well as previous network weights can be derived using backpropagation and is not discussed further here.

There are several ways to decode an utterance given the model. The simplest decoder is called greedy decoder [52], which basically performs $argmax(y_k^t)$ over the label set L' for all time step t. After obtaining the labels for each time step, the many-to-one mapping B is used to get rid of blanks and squash repeated alphabet symbols. However, greedy decoding provides no guarantee that the decoding is optimal. Other CTC decoding methods include prefix-search [54], beam search [55], and WFST-based decoding [92], some of which, during decoding, uses a lexicon and/or a language model [55, 92] to further improve phone error rate/word error rate.

One model that uses the CTC criterion is the Deep Speech 2 model [1]. In fact, the Bi-LSTM model trained in this thesis is directly modified from the Deep Speech 2 model, just to constrain the total number of parameters. In this section, the author will only review the model related part of Deep Speech 2 (DS2).

DS2 takes a spectrogram of power normalized audio clips as input features. It then goes through two layers of convolution in both the time and frequency axis (to model both local temporal invariance and spectral variance), each of which is followed by a clipped ReLU function. Usually, in the first convolutional layer, the time dimension is reduced via striding. Following the convolutional layers are stacked bidirectional recurrent layers, with the activation from the forward unit and the backward unit summed before going into the next layer. Upon reaching the last layer, it goes through a softmax output layer that computes a probability for each of the possible outputs. The outputs of the English model in DS2 includes English characters, space, apostrophe and blank symbol for CTC (note that the modified model in this thesis does not use Dutch/English characters but instead IPA phones). As mentioned earlier, the model is trained using CTC loss, which learns a probability distribution over all label sequences.

DS2 incorporates several techniques for improving the model design. First, they applied a special type for Batch Normalization called Sequence-Wise Batch-Norm [75]. A normal Batch-Norm [61] operation is defined as an operation to transform the layer output by

$$B(x) = \gamma \frac{x - E[x]}{(Var[x] + \epsilon)^{1/2}} + \beta \quad (60)$$

where the mean and variance are taken as the empirical mean and variance, and γ and β are learnable parameters for scaling and shifting, respectively, before feeding into a non-linear activation function. Sequence-wise Batch-Norm computes the mean and variance both over all items in the minibatch and over the length of the input, and in terms of RNN can be defined as:

$$\vec{h}_t^l = f(B(W^l h_t^{\vec{l-1}}) + U^l \vec{h}_{t-1}^l) \quad (61)$$

The DS2 paper has found out that as much as 12% of improvement could be achieved had this type of Batch-Norm is used.

Another technique is called Sorta-Grad, which deals with varying length sequences during training. Recall that the CTC loss function is defined as

$$L(x; y; \theta) = -\log \sum_{\pi \in B^{-1}(\mathbf{l})} \prod_{t=1}^{T} y_{\pi_t}^t \quad (62)$$

where π_l is the network output at time t, and $y_{\pi_t}^t$ is its probability output from the network. As the product term shrinks with a larger T, DS2 paper argues that the length of the utterance could be used as a heuristic for difficulty, and thus in the first training epoch, should feed the model in increasing order of length.

One last technique worth mentioning is called row convolution, which is applied to unidirectional variants of DS2 models to reach the same level of performance as bidirectional ones. It assumes that at every time step, a future context matrix

$$H = h_{t:t+\tau} = [h_t, h_{t+1}, ..., h_{t+\tau}] \quad (63)$$

is used, and thus defines a parameter matrix W of the same size $d \times (\tau + 1)$. The output of applying W to H is defined as

$$r_{t,i} = \sum_{j=1}^{\tau+1} W_{i,j} h_{t+j-1,i} \quad \text{for} \quad 1 \leq i \leq d \quad (64)$$

By placing the row convolution above all recurrent layers, the paper claimed to have gotten an even better character error rate than the best bidirectional model on Mandarin data.

2.4 Visualization

Weight visualization of deep neural networks has been a relatively open and active topic across many fields [4, 125]. In this thesis, the author used the same visualization scheme as in [113] to investigate how the decision boundary and clustering behaviors (w.r.t. natural /l/ sounds, natural /r/ sounds and ambiguous [l/r] sounds) change during the time-course of perceptual learning. This visualization scheme first uses a variant of Non-negative Factor Analysis (NFA) [3] for GMM weight decomposition. After necessary normalization, it tries to model the DNN activations for a given phone utterance **v** as a shift from the mean activation **m**. The shift itself is modeled as the product of a fat matrix **T** and a low-dimensional summary vector for the given utterance **w**, which in effect captures the most important non-negative variability of the DNN activations [113]. In this step, the matrix **T** is optimized using all sounds from the phone set in the retuning set, plus the ambiguous sound.

Following the above step, the summary vectors **w** for phones of interest (i.e. /l/, /r/ and [l/r]) are extracted and projected onto the first three principal axes with the greatest variance using the well-known algorithm called Principal Component Analysis (PCA) [13, 60, 63, 100]. After this

step, the dimensionality of the summary vectors for the sounds of interest are further reduced and could be plotted in a common 3D space, as defined by the first three principal axes.

In the following sections, Principal Component Analysis will first be reviewed, followed by Non-negative Factor Analysis.

2.4.1 Principal Component Analysis

Principal Component Analysis (PCA) is widely used for dimensionality reduction, lossy data compression, feature extraction, and data visualization [63]. The derivation below came from [13].

PCA could be formulated as two different problems, with the first being maximum variance formulation and the second being minimum error formulation:

A. Maximum Variance Formulation [13, 60]
Suppose a D-dimensional dataset $\{\mathbf{x}_n\}$ of size N needs to be projected onto an M-dimensional space where $M < D$. It is apparent that such a projection needs to capture as much variance of the original dataset as possible.

Suppose M = 1, and the direction of projection is specified by the vector $\mathbf{u}_1$ in the D-dimensional space. Without loss of generality, also suppose

$$\mathbf{u}_1^T \mathbf{u}_1 = 1 \quad (65)$$

Therefore, each data point can be projected as a scalar $\mathbf{u}_1^T \mathbf{x}_n$, and the variance could be calculated as

$$\frac{1}{N} \sum_{n=1}^{N} \{\mathbf{u}_1^T \mathbf{x_n} - \mathbf{u}_1^T \bar{\mathbf{x}}\}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad (66)$$

where

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \quad (67)$$

Maximizing $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with $\mathbf{u}_1^T \mathbf{u}_1 = 1$ can be done using Lagrange multiplier as the unconstrained maximization of

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1) \quad (68)$$

and setting the derivative w.r.t $\mathbf{u}_1$ to zero yields

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad (69)$$

so $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ with eigenvalue λ_1. Left multiplying by $\mathbf{u}_1^T$ yields

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \quad (70)$$

and so $\mathbf{u}_1$ needs to be the eigenvector with the largest eigenvalue to maximize the variance on the first projected axis.

For M > 1, the principal axes could be incrementally chosen as the eigenvector with the M-th largest eigenvalue λ_M in order to maximize total capture variance. To prove this is true, suppose this holds for $\{\mathbf{u}_1 ... \mathbf{u}_{M-1}\}$, and now $\mathbf{u}_M$ needs to be determined. It is obvious that this new vector needs to be orthogonal to the previous principal axes, and so

$$\mathbf{u}_M^T \mathbf{u}_s = 0 \ \ for \quad s = \{1...M-1\} \quad (71)$$

It is also obvious that the added variance $\mathbf{u}_M^T \mathbf{S} \mathbf{u}_M$ needs to be maximized in order for the total captured variance to be maximized (as the invariant states that $\mathbf{u}_1 .. \mathbf{u}_{M-1}$ captures the maximum possible variance for an (M - 1) - dimensional space). Therefore, using Lagrange multipliers, we have

$$\mathbf{u}_M^T \mathbf{S} \mathbf{u}_M + \lambda_M (1 - \mathbf{u}_M^T \mathbf{u}_M) + \textstyle\sum_{i=1}^{M-1} \eta_i \mathbf{u}_M^T \mathbf{u}_i \quad (72)$$

Setting the derivative w.r.t. $\mathbf{u_M}$ to zero yields

$$0 = 2\mathbf{S}\mathbf{u}_M - 2\lambda_M \mathbf{u}_M + \sum_{i=1}^{M-1} \eta_i \mathbf{u}_i \quad (73)$$

Again, moving the middle term to the left and using the orthogonal constraint gives

$$\mathbf{S}\mathbf{u}_M = \lambda_M \mathbf{u}_M \quad (74)$$

Left multiplying by $\mathbf{u}_M^T$ shows that the maximum value is reached by choosing $\mathbf{u}_M$ to be the eigenvector that corresponds to the M-th largest eigenvalue.

B. Minimum-error formulation [13, 100]

PCA can also be formulated to minimize the projection error. Suppose again the dataset $\{\mathbf{x}_n\}$ is of D-dimension. Further, suppose there is an orthogonal set of basis vectors $\{\mathbf{u}_i\}, i = 1, ..., D$ s.t.

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \quad (75)$$

Using this basis,

$$\mathbf{x}_n = \textstyle\sum_{i=1}^{D} \alpha_{ni} \mathbf{u}_i = \sum_{i=1}^{D} (\mathbf{x_n}^T \mathbf{u}_i) \mathbf{u}_i \quad (76)$$

with the last one using

$$\alpha_{nj} = \mathbf{x}_n^T \mathbf{u}_j \quad (77)$$

Suppose the M-dimensional linear subspace (with minimum projection error) is represented by the first M basis vectors. Then the approximation in the M-dimensional space for each $\mathbf{x}_n$ is

$$\hat{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni} \mathbf{u}_i + \sum_{i=M+1}^{D} b_i \mathbf{u}_i \quad (78)$$

and the loss becomes:

$$J = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} ||\mathbf{x}_n - \hat{\mathbf{x}}_n||^2 \quad (79)$$

Taking the derivative w.r.t. z_{ni} and b_i and setting to zero gives

$$z_{ni} = \mathbf{x}_n^T \mathbf{u}_i \quad (80)$$

and

$$b_i = \bar{\mathbf{x}}^T \mathbf{u}_i \quad (81)$$

Therefore

$$\mathbf{x}_n - \hat{\mathbf{x}}_n = \sum_{i=M+1}^{D} \{(\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{u}_i\} \mathbf{u}_i \quad (82)$$

and the distortion J is now

$$J = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \sum_{i=M+1}^{D} (\mathbf{x}_n^T \mathbf{u}_i - \bar{\mathbf{x}}^T \mathbf{u}_i)^2 = \sum_{i=M+1}^{D} \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i \quad (83)$$

The solution to minimizing J above is achieved by choosing $\mathbf{u}_i, i = 1, ..., D$ as the eigenvectors of the covariance matrix $\mathbf{S}$ s.t the i-th eigenvector has the i-th largest eigenvalue λ_i, and so the minimal value of J is simply the sum of the $(D - M)$ smallest eigenvalues.

2.4.2 Non-negative Factor Analysis

Non-negative Factor Analysis (NFA) was first proposed in [100] as a subspace method for GMM weight adaptation, which provided complementary information to GMM mean adaptation (for example, the i-vector framework [26]) for language/dialect recognition. The derivations below came from the original NFA paper [3].

The first concept of NFA is the notion of a Universal Background Model (UBM) [108]. UBM assumes that the utterance matrix:

$$\mathbb{X} = \{x_1, ..., x_t, ..., x_\tau\} \quad (84)$$

follows the likelihood function:

$$p(x_t|\lambda) = \sum_{c=1}^{C} b_c p(x_t|\mu_c, \Sigma_c), \lambda = \{b_c, \mu_c, \Sigma_c\}, c = 1, ...C \quad (85)$$

with acoustic vectors x_t and parameters of the GMM specified by λ. GMM weight adaptation, therefore, attempts to adapt the UBM weights b_c to utterance-dependent weights w_c. The utility function for such weight adaptation resembles the auxiliary function in the E-M algorithm [14] for estimating GMM:

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c p(x_t|\mu_c, \Sigma_c) \quad (86)$$

with $\gamma_{c,t}$ being the posterior count of the c-th mixture and is held constant during the optimization. Because the Gaussian pdfs remain unchanged during optimization, the above utility can be further simplified as

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c \quad (87)$$

NFA further assumes that, for a given utterance, each of the w_c could be decomposed as a univariate shift from the UBM weight b_c, and therefore has the form

$$w_c = b_c + \mathbf{L_c r} \quad (88)$$

where $\mathbf{L_c}$ is the c-th row of a subspace matrix $\mathbf{L}$ of size $C \times p (p \ll C)$, and $\mathbf{r}$ being a summary vector for the utterance that best describes all the (univariable) shifts. The difference between NFA and the more well-known NMF [78] (non-negative matrix factorization) is that the entries of the matrix $\mathbf{L}$ and the summary vectors $\mathbf{r}$ are allowed to be negative for NFA, as long as the entries in $\mathbf{w}$ are not.

Finding the joint subspace matrix $\mathbf{L}$ and the individual summary vectors $\mathbf{r}$ involves a two−step iterative optimization similar to E−M. In the first step, $\mathbf{L}$ is held constant and all $\mathbf{r}$ are updated. In the second step, all summary vectors $\mathbf{r}$ for all the utterances are held constant and the shared subspace matrix $\mathbf{L}$ is updated. The equations corresponding to the two steps in each iteration is listed as below:

First step: Updating $\mathbf{r}$ for all utterances

Using the assumptions above, the utility function for a single utterance can be rewritten as:

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log\left(b_c + \mathbf{L_c r}\right) = \bar{\gamma}'(\mathbf{X}) \log\left(\mathbf{b} + \mathbf{L r}\right) \quad (89)$$

where

$$\bar{\gamma}(\mathbf{X}) = \sum_t [\gamma_{1,t} .... \gamma_{C,t}]' \quad (90)$$

and

$$\mathbf{b} = [b_1 ... b_C]' \quad (91)$$

Given the non-negative constraint on all the adapted weights w_c, the problem becomes:

$$\max_{\mathbf{r}} \Phi(\lambda, \mathbf{r}) \quad (92)$$

Subject to

$$\mathbf{1}(\mathbf{b} + \mathbf{L r}) = 1$$
$$\mathbf{b} + \mathbf{L r} > 0$$

From the equality constraint, we can get $\mathbf{1 L r} = 0$. Here, assume that another constraint $\mathbf{1 L} = 0$ is enforced(in the second step), and so the equality constraint disappears as it holds for all $\mathbf{r}$. The inequality constraint is satisfied by carefully controlling the step size α_E of update. With constraints relaxed, maximizing the utility w.r.t. r becomes the following iterative update:

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \alpha_E \nabla \Phi(\lambda, \mathbf{r}_{i-1}) \quad (93)$$

where

$$\nabla \Phi(\lambda, \mathbf{r}) = \mathbf{L}' \frac{[\bar{\gamma}'(\mathbf{X})]}{[\mathbf{b} + \mathbf{L r}(\mathbf{X})]} \quad (94)$$

To obtain the initial values for $\mathbf{r}$, the following equation is used:

$$\mathbf{r}_{pinv} = \mathbf{L}^{\dagger} [\tfrac{1}{\tau} \bar{\gamma}(\mathbf{X}) - \mathbf{b}] \quad (95)$$

and $\mathbf{r}_0$ is chosen as $\theta \mathbf{r}_{pinv}$ where θ is iteratively halved until it satisfies the inequality constraint.

Second Step: Updating subspace matrix $\mathbf{L}$ jointly

As the subspace matrix is shared among all utterances, the utility function $\tilde{\Phi}(\lambda, \mathbf{L})$ for updating $\mathbf{L}$ is the summation of the utility functions $\Phi(\lambda, \mathbf{r}_s)$ for all utterance s. Therefore, the problem becomes:

$$\max_{\mathbf{L}} \tilde{\Phi}(\lambda, \mathbf{L}) \quad (96)$$

Subject to

$$\mathbf{1}(\mathbf{b} + \mathbf{L r}(\mathbf{X}_s)) = 1$$
$$\mathbf{b} + \mathbf{L r}(\mathbf{X}_s) > 0$$
$$s = 1, ..., S$$

where

$$\tilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \bar{\gamma}'(\mathbf{X}_s) \log\left(\mathbf{b} + \mathbf{L r}(\mathbf{X}_s)\right) \quad (97)$$

As the condition $\mathbf{1L}$ is used in the first step to relax the equality constraints there, to update $\mathbf{L}$, projected gradient descent needs to be used:

$$\mathbf{L}_i = \mathbf{L}_{i-1} + \alpha_M \mathbf{P} \nabla \tilde{\Phi}(\lambda, \mathbf{L}_{i-1}) \quad (98)$$

where

$$\nabla \tilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \frac{[\bar{\gamma}(\mathbf{X}_s)]}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathbf{X}_s)]} \mathbf{r}'(\mathbf{X}_s) \quad (99)$$

and

$$\mathbf{P} = \mathbf{I} - \tfrac{1}{C} \mathbf{1}' \mathbf{1} \quad (100)$$

To initialize $\mathbf{L}$, Principal Component Analysis is used on the matrix formed by the ML estimates of $\mathbf{w}$ from all utterances.

3. Methodology

3.1 Investigating the Time Course of DNN Perceptual Learning

To mimic the set-up for the human listener experiment, we first trained a DNN on a Dutch speech corpus. To mimic or create a Dutch listener, we first trained a baseline DNN using read speech from the Spoken Dutch Corpus (CGN; [97]). The read speech part of the CGN consists of 551,624 words spoken by 324 unique speakers for a total duration of approximately 64 hours of speech. A forced alignment of the speech material was obtained using a standard Kaldi [105] recipe found online [50]. The speech signal was parameterized using a 64-dimensional vector of log Mel spectral coefficients with a context window of 11 frames, each having a segment length of 25 ms with a 10 ms shift between frames. Per-utterance mean-variance normalization was applied. The CGN training data were split into a training (80% of the full data set), a validation (10%), and a test set (10%) with no overlap in speakers.

We used a simple fully-connected, feed-forward network with five hidden layers, 1024 nodes per layer, with logistic sigmoid nonlinearities as well as batch-normalization and dropout after each layer activation. The output layer was a softmax layer of size 38, corresponding to the number of phonemes that existed in our training labels. The model was trained on CGN for 10 epochs using an Adam optimizer with a learning rate of 0.001. After 10 epochs, we reached a training accuracy of 85% and a validation accuracy of 77% on CGN.

Because we aimed to investigate the DNN's ability to serve as a model of human perceptual learning, we used the same acoustic stimuli as used in the human perception experiment [112] for retraining the DNN (also referred to as retuning). The retraining material consisted of 200 Dutch words produced by a female Dutch speaker in isolation: 40 words with final [ɹ], 40 words with final [l], and 120 'distractor' words with no [l] and [ɹ]. For the 40 [l]-final words and the 40 [ɹ]-final words, versions also existed in which the final [l] or [ɹ] was replaced by the ambiguous [l/ɹ] sound. Forced alignments were obtained using a forced aligner for Dutch from the Radboud University. For four words no forced alignment was obtained, leaving 196 words for the experiment.

To mimic the two listener groups from the human perceptual experiment, and to mimic a third group with no exposure to the ambiguous sound (i.e., a baseline group), we used three different configurations of the retuning set:

Amb(iguous)L model: trained on the 118 distractor words, the 39 [ɹ]-final words, and the 39 [l]-final words in which the [l] was replaced by the ambiguous [l/ɹ].

Amb(iguous)R model: trained on the 118 distractor words, the 39 [l]-final words, and the 39 [ɹ]-final words in which the [ɹ] was replaced by the ambiguous [l/ɹ].

Baseline model: trained on all 196 natural words (no ambiguous sounds). This allows us to separate the effects of retuning with versus without the ambiguous sounds.

In order to investigate the time-course of phoneme category adaptation in the DNNs, we used the following procedure. First, the 196 words in the three retuning sets were split into 10 bins of 20 distinct words, except for the last two bins, which each contained only 18 words. In order to be able to compare between the different retuning conditions, the word-to-bin assignments were tied among the three retuning conditions. Each word appeared in only one bin. Each bin contained: 4 words with final [r] (last bin: 3 words) + 4 words with final [l] (penultimate bin: 3 words) + 12 'distractor' words with no [l] or [r] (last two bins: 11 words). The difference between the retuning conditions is:

AmbL: the final [l] in the 4 [l]-final words were replaced by the ambiguous [l/ɹ] sound.

AmbR: the final [ɹ] in the 4 [ɹ]-final words were replaced by the ambiguous [l/ɹ] sound.

Baseline: only natural words.

The [l]-final, [ɹ]-final, and [l/ɹ]-final sounds of the words in bin t from all three retuning sets, combined, functioned as the test set to bin t-1. As all the acoustic signals from the test bin were unseen during training at the current time step, we denote this as "open set evaluation". Figure 1 explains incremental adaption. Note that the final bin was only used for testing; because at t=10, there is no subsequent bin that could be used for testing.

Retuning was repeated five times, with five different random seeds for permutation of data within each bin, for each retuning condition/model. Each time, for every time step of incremental adaptation, we retrained the baseline CGN-only model using bin 0 up to bin t-1 of the retraining data for 30 epochs using an Adam Optimizer with a learning rate of 0.0005. The re-tuning accuracy on the training set after 30 epochs always reached an accuracy of 97.5 – 99%.

```
for each retuning set from {Baseline, AmbL, AmbR}
    Test the CGN-only model using bin 0 from the test set

    for t in [1,9]:
        Retrain the CGN-only model using bin 0 up to bin t-1
        Test the retrained model from bins 0 through t-1 using test set bin t
```

Fig. 1. Incremental retuning procedure for the open set evaluation.

We then carried out four experiments aiming at different angles of the retuning process.

In the first experiment, we investigated the amount of training material needed for perceptual learning in a DNN to occur. Therefore, for each of the three retuning set (Baseline, AmbL, AmbR), we plotted out the 9-step classification rate on the test set, with x-axis being the time step and y-axis being the percentage of the three sound classes (natural /l/, natural /r/, and ambiguous [l/r]) classified as either /l/ or /r/ by the DNN). By comparing the plot of AmbL and AmbR sets with the Baseline set, we could figure out at which time step perceptual learning actually occurs, and as described earlier, each time step consisted of an increasing number of retuning tokens, the amount of training material needed could be determined.

As we had found out that the classification rates made a significant jump after just the first bin, in the second experiment, we further investigated how the pre-trained DNN adapted to this very first training bin, which consisted of very limited re-tuning data. To do this, we evaluated the classification rates by training the CGN-only model using the first training bin (training bin 0) from each experiment set (natural, AmbL, AmbR) for 30 epochs, and recorded the percentage of [l], [ɹ], and ambiguous [l/ɹ] sounds from the second test bin (test bin 1) that were classified as either [l] or [ɹ] before the first epoch (t=0), and after each epoch of training ($1 \leq t \leq 30$).

In the third experiment, we investigated where the retuning takes place. We did so by plotting out the inter-category distance ratio metric as proposed in [113], for all the five hidden layers of the DNN model, during the 9-step of incremental retuning. The measure quantified the degree to which lexical retuning has modified the feature representations at the hidden layers using a single number. First, the 1024-dimensional vector of hidden layer activations was re-normalized, so that each vector summed to one. Second, the Euclidean distances between each [l/ɹ] sound and each [l] sound were computed, after which the distances were averaged over all [l/ɹ]-[l] token pairs, resulting in the average [l]-to-[l/ɹ] distance. Third, using the same procedure, the average [ɹ]-to-[l/ɹ] distance was computed. The inter-category measure was then the ratio of these two distances.

To visualize the adaptation course, we chose to use the same DNN weight visualization scheme as in [113]. This visualization scheme is based on Non-Negative Factor Analysis (NFA) [3], which was first proposed for Gaussian Mixture Model (GMM) weight adaptation, followed by Principal Component Analysis (PCA) [13, 60, 63, 100]. In our DNN model, every dense layer was followed by a sigmoid activation layer, which squashed the output into the range of (0,1). The values from this 1024-dimensional post-sigmoid vector were still not directly interpretable as GMM weights, so in a subsequent normalization step, we calculated the L1-norm of the vector and divided every entry of that vector by this L1-norm. After this step, all the entries in the activation vector summed up to one and could be treated as GMM weights in the NFA algorithm. The normalized activation matrices for all the phonetic segments from all the time steps were first fed into the NFA algorithm, and the extracted summary vectors of /l/, /r/ and [l/r] were then fed into PCA for visualization in 3D space (as defined by the first three principal axes).

3.2 Investigating the Time Course of Bi-LSTM Perceptual Learning

To create a baseline Dutch listener, we again trained the Bi-LSTM on the 64-hour read speech section of the CGN corpus [97]. As CTC loss does not need any alignment during training, the phonetic transcriptions of the segmented utterances from CGN were used as ground truth during training. Raw spectrograms with a segment length of 25ms and a shift of 10ms between frames were used as input, and per-utterance mean-variance normalization was applied on the spectrograms before feeding into the model. The ASR model was modified from Baidu's DeepSpeech2 [1] to constrain the total number of parameters, and consisted of two layers of 2D-convolution on the spectrogram (with the same parameter settings as in DeepSpeech2), followed by 6 layers of batch-normalized, bi-directional LSTM layers (with dimension greatly reduced), followed by a dense layer that was shared across time, and a softmax output layer for the CGN phone set. The phone error rate of this model on the CGN test set was ~12% after 13 epochs, using a simple greedy decoder.

The retraining material was the same as that used in the previous section, i.e., three sets of 200 words, with 120 distractors, 40 /l/-final words (with one set containing ambiguous sounds), and 40 /r/-final words (with another set containing ambiguous sounds). To investigate the time-course, every retuning set is again split into 10 bins of 20 words, with 4 /l/-final words, 4 /r/-final words, and 12 natural words. As forced alignments were not needed for this model, all 200 words from the retuning set were used. The two ambiguous models were again trained by replacing the corresponding natural /l/-final (/r/-final) words to words with ambiguous [l/r], and the open test set for all three models during each time-step was again chosen as the /l/-final and /r/-final words from all three sets. Retuning was repeated 5 times to reduce noise, with 13 epochs per step. The incremental training procedure was exactly the same as in Figure 1.

In this part, only experiments one and four were repeated, with slight modifications to their procedures so as to better fit with the new model.
For the modified experiment one, we again investigated the amount of training material needed for perceptual learning in the new Bi-LSTM model to occur. As the new model output a phonetic transcription for the entire utterance, the percentages of natural /l/, natural /r/ and ambiguous [l/r] classified as /l/ or /r/ were calculated as follows:

1. For each /l/-final word or /r/-final word (with either ambiguous or natural sound) in the current test set, feed it through the model to get its phonetic transcription.
   If the last output phone in the transcription is a /l/, increment the /l/ count for the corresponding class (out of natural /l/, natural /r/, ambiguous [l/r)]; otherwise, increment the /r/ count
2. Average the /l/ and /r/ counts within its corresponding sound class for the current time-step. For natural /l/ and /r/ class, the dividend was 4 for every time step, as there were 4 natural /l/-final words and 4 natural /r/-final words; for ambiguous [l/r], the dividend was 8 as each /l/-final and /r/-final word had an ambiguous counterpart)

For the modified experiment four, we again tried to visualize the time course of adaptation. Like experiment four described in the previous section, we first applied NFA to find a summary vector for each of the phone segments within our 280-word adaptation set, extracted out the summary vectors of natural /l/, natural /r/ and ambiguous [l/r] segments, and plotted their projection onto the first three principal axes. As no forced alignment is available/used in this part, we used the phonetic segments implicitly aligned by the CTC output function. For example, suppose at frame number 7 the CTC decoder started to output the phone /i/ and at frame 14 (after consecutive outputs of /i/, possibly followed by blanks), it switched to phone /r/, then frame 7 to frame 13 were considered as a segment for phone /i/. We again only visualized the final hidden LSTM layer activation vectors (sigmoid activation) as we believed that this final hidden layer encoded the most information with respect to phone boundaries.

3.3 Modeling the Dutch ASR Model as a Second Language Learner

In order to further compare the learning behaviors of a deep neural network-based ASR with those of a human, in this part, we tried to adapt a well-trained Bi-LSTM model on a first language (in this case, Dutch) to a second language (in this case, English), and see if specific techniques could be used to improve the performance of the cross-lingual model.

The model architecture used was exactly the same as the one described in 3.2: it was modified from Deep Speech 2 architecture and consisted of 2 convolutional layers on the normalized spectrogram, followed by 6 layers of Bi-LSTM. Sequence-wise batch normalization was used between consecutive layers of the LSTM layer. The final output first went through a fully-connected layer that was shared across time and then through a softmax layer which calculated the probability distribution over the label class. Again, a Dutch model was pre-trained before the adaptation experiments.

For adapting to a new language (English), the Flickr 8k Audio Caption Corpus [56] was used. It consists of 40,000 spoken captions of 8,000 natural images from the Flickr8k image-caption dataset. It was collected in 2015 to investigate multimodal learning schemes for unsupervised speech pattern discovery. Here to investigate the early stage of second language learning for a neural network-based ASR, only ~3600 utterances were selected out of all the spoken captions.

The adaptation scheme used here was a bit different from what had been used in the previous two sections, however. In the previous two sections, ground truth labels of the adaptation set (either phone label for the DNN model or complete phonetic transcriptions for the Bi-LSTM model) were provided to the model to supervise the adaptation. Here, as Dutch and English share a similar phone set (and considered as two somewhat related languages), we applied the self-training scheme as in [110]. In this scheme, the L1 model was first initialized using linguistic knowledge: the missing English diphthongs were split in half, with each half corresponding to an existing Dutch phone in the Dutch phone set; new softmax layer vectors were initialized for the rest of the missing phones, as follows:

$$\vec{V}_{|\varphi|,L2} = \vec{V}_{|\varphi|,L1:1} + 0.5(\vec{V}_{|\varphi|,L1:2} - \vec{V}_{|\varphi|,L1:3}) \quad (101)$$

where $\vec{V}_{|\varphi|,L2}$ was the new softmax layer vector for the missing English phone, and $\vec{V}_{|\varphi|,L1:1}$, $\vec{V}_{|\varphi|,L1:2}$, and $\vec{V}_{|\varphi|,L1:3}$ were the three corresponding Dutch phones chosen via linguistic knowledge about the two languages. After the above initialization, the model was then asked to directly transcribe L2 utterances using self-labeling, and 70% of the utterances with a lower self-labeled phone error rate (as determined using ground truth transcriptions) were chosen for subsequent self-adaptation. Except for calculating the phone error rates, the ground truth transcriptions were otherwise not used.

In the first step, some further analyses of the results were completed. Like in the previous experiments, we tried to determine if the 2%-3% decrease in error rate was step-like or gradual across the second adaptation step and if it was step-like, at which point in time it occurred. To do this, the chosen utterances were split into 10 bins, and during self-adaptation, were incrementally fed to the model. Other analyses include calculating the recognition rates on all the L2 phones before and after the adaptation step, as an attempt to figure out where the slight decrease in error rate came from (i.e., was it because the recognition rates on the missing phones improved after self-adaptation; or was it because the recognition rates were higher on the shared phones; or was it hard to tell?) and what phones contributed most to the still relatively large error rate after self-adaptation. To do this, the recognition rates were plotted for all the phones across 10 steps as a bar plot and some phones of interest (such as the missing L2 phones) were extracted out for further examination.

As will be discussed in further detail in the next chapter, the results from the first step were not very promising: the model failed to adapt on all of the missing L2 (English) diphthong, as well as on almost all of the rest of the missing phones that were initialized via a linear combination of three L1 (Dutch) phones. At this point, we felt that if further improvements to the error rate were to be achieved, we would need to find ways to improve the acoustic model so that those missing L2 phones could be better adapted. Therefore, two prospective methods on further improving the acoustic model of this self-training paradigm were pursued:

The first method involved multi-task learning of both the phones and the articulatory feature equivalent class corresponding to each phone from the raw speech input. For consonants, the equivalence class was defined by place, manner, and voicing; for vowels, the equivalent class was defined by rounding, height, and frontness. The goal was that, by forcing the L1 model to learn language-agnostic, sub-phonetic features from the raw speech input, as well as an equivalent mapping from articulatory features to phones (and vice versa), when adapted to a second language, the model would already have some capability of associating sets of articulatory features with unseen foreign speech input. Because articulatory features are language agnostic, we speculated that the error rate for directly transcribing articulatory features after switching to a new language would be significantly lower than transcribing the phones directly, and therefore self-training on the articulatory features to update model weights and re-transcribing the phones again before a subsequent phone adaptation step could potentially yield better results, if the same overall paradigm was kept fixed.

To do this, six additional sets of transcriptions containing the labeling for each of the articulatory feature class were obtained using a phone to articulatory feature mapping for Dutch, for each of the training/test/validation utterances from CGN corpus. We then trained an updated version of the L1 (Dutch) listener model, with 6 additional softmax layers appended to the output of the last LSTM layer, each of which corresponded to one of the six types of articulatory features. The weights prior to the softmax (i.e., those of convolutional layers and stacked Bi-LSTM layers) were shared among all tasks. The model was trained for a total of 14 epochs, using Adam Optimizer with a learning rate of 8e-4 for the phone class and 8e-3 for all the articulatory features.

After training the updated Dutch listener model and initializing the missing phones and missing articulatory feature (English contained an extra label for the place articulatory feature class called dental, and its softmax layer vector was initialized as the mean of those for labiodental and alveolar), the articulatory feature transcriptions and phonetic transcriptions on the English adaptation set were obtained (i.e., via self-labeling). We then carried out an incremental update scheme to the acoustic model weights, as follows: in the first step, we chose one class of articulatory feature out of the rest of that have yet to be adapted, picked a certain percentage of the utterances that scored the lowest token error rates, and trained on the self-labeled transcriptions obtained in a previous step; in the second step, we re-transcribed the whole set of utterances (for both the phonetic transcriptions and articulatory feature transcriptions) using the new model weights. This was repeated on all six classes of articulatory features until no further improvement could be gained. The model self-trained on articulatory features was then used to generate phonetic transcriptions for all the utterances, and a certain percentage of them were chosen to self-adapt the model.

However, as will be mentioned later, the model still did not give us too much of an improvement. Therefore, we tried to gain ideas from a traditional second language classroom setting: when learners first approached a second language, they would usually be taught how to recognize simple, isolated words before moving onto longer utterances. Also, according to the loss function of CTC, shorter utterances would be preferred at the early stage of network learning (which was why SortaGrad was used in Deep Speech 2). Therefore, after initializing the English ASR model, we first tried to adapt it to a separate isolated word set called TI46 word speech database [81], which is a corpus that contains 16 speakers (8 male and 8 female). Each speaker spoke a total of 46 simple words (e.g. zero to nine and A to Z). As these words are relatively short, we hoped that by self-adapting to this isolated word set, prior to self-adapting to those longer utterances in Flickr-8k, the model would be able to better detect phone boundaries.

However, this approach basically failed as the recording condition of TI 46 word set seemed too different from that of CGN or Flickr-8k, and the phone error rate (around 95%) before adaptation was deemed too high for self-training to be successful.

We then tried to segment out individual words for the utterances in the Flickr-8k corpus. However, the error rate was still too high (90%) for any type of self-training to be successful.

Carefully listening to the segmented words suggested that as those words were segmented from continuous speech, the starting and ending sounds seemed so unnatural that it was even very hard for a human being to correctly recognize them.

Up to the writing of this thesis, how to effectively carry out the second proposed approach remained unsolved.

4. Experimental Results

4.1 Classification Rates For DNN Perceptual Learning

In the first experiment, we investigated the amount of training material needed for perceptual learning in a DNN to occur. Classification accuracy was computed for all frames, but since we are primarily interested in the [l], [ɹ], and the ambiguous [l/ɹ] sound, we only report those. Figures 2 through 4 show the proportion of correct frame classifications as solid lines, i.e., [l] frames correctly classified as [l] and [ɹ] frames correctly classified as [ɹ], for each of the 10 bins ($0 \leq t \leq 9$). Dashed lines show, for example, the proportion of [l] frames incorrectly classified as [ɹ], and of [ɹ] frames incorrectly classified as [l]; the rate of substitutions by any other phone is equal to 1.0 minus the solid line minus the dashed line. The interesting case is the classification of the [l/ɹ] sound (see triangles), which is shown with a dashed line when classified as [l] and with a solid line when classified as [ɹ]. Note, in the legend, the capital letter denotes the correct response, lowercase denotes the classifier output, thus, e.g., L_r is the percentage of [l] tokens classified as [ɹ].

Figure 2 shows the results for the baseline model retrained with the natural stimuli. The baseline model shows high accuracy in the classification of [ɹ]. The [l] sound is classified with high accuracy at t=2, then drops for increasing t, up to t=8. The [ɹ] sound, on the other hand, is classified with very high accuracy after seeing a single bin of retuning data, with very little further improvement for subsequent bins. The [l/ɹ] sound (not part of the training data for this model) is classified as [ɹ] about 70% of the time, and as [l] about 10% of the time, with the remaining 20% of instances classified to some other phoneme.

Figure 3 shows the results for the model retrained with the ambiguous sound bearing the label of /l/. The AmbL model has a high accuracy in the classification of the [ɹ]; however, the accuracy of natural [l] is less than 50% after the first bin and continues to worsen as more training material is added. The lexical retuning dataset contains no labeled examples of a natural [l]; apparently, in this case, the model has learned the retuning data so well that it forgets what a natural [l] sounds like.

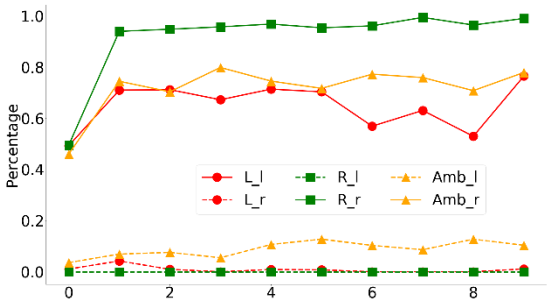


Fig. 2. Proportion of [l] and [ɹ] responses by the baseline model, retrained with natural stimuli, per bin.

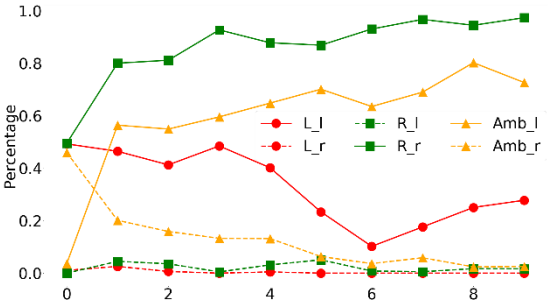


Fig. 3. Proportion of [l] and [ɹ] responses by the AmbL model, retrained with [l/ɹ] labeled as [l], per bin.

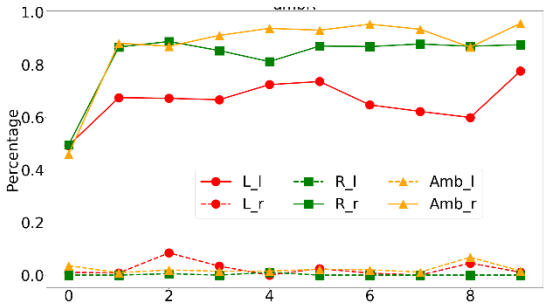


Fig. 4. Proportion of [l] and [ɹ] responses by the AmbR model, retrained with [l/ɹ] labeled as [ɹ], per bin.

Importantly, after the first bin, the network has correctly learned to label the [l/ɹ] sounds as [l], indicating 'perceptual learning' by the AmbL system. The classification of [l/ɹ] as [l] continues to rise slightly for subsequent bins. While the AmbL model already correctly recognizes most [l/ɹ] sounds as [l] after the first bin, recognition further improves for subsequent bins.

Figure 4 shows the results for the model retrained with the ambiguous sound labeled as /r/. The AmbR model has high accuracy for both the [l] and [ɹ] sounds. So, unlike the AmbL model, the AmbR model did not forget what a natural [ɹ] sounds like. Moreover, after the first bin, this model has learned to classify [l/ɹ] as [ɹ] more than 85% of the time, which is a 10% increase over the

model trained on the natural sounds at the same time step, thus showing perceptual learning. Unlike the AmbL model, additional [l/ɹ] training examples show little tendency to further increase the classification of [l/ɹ] as [ɹ], up to and including the last point.

Interestingly, all phones, including the natural [l] and [ɹ] as well as the ambiguous phone, show classification accuracy of around 50% prior to retraining. This rather low accuracy is most likely due to the differences in recording conditions and speakers between the CGN training set and the retraining sets. After retraining with the first bin, the classification accuracies make a jump in all models, with little further adaptation for subsequent bins, although the AmbL shows a small increase in adaptation for later bins, while this is not the case for the baseline and AmbR models. This adaptation suggests that the neural network treats the ambiguous [l/ɹ] exactly as it treats every other difference between the CGN and the adaptation data: In other words, exactly as it treats any other type of inter-speaker variability. In all three cases, the model learns to correctly classify test tokens after exposure to only one adaptation bin (only 4 examples, each, of the test speaker's productions of [l], [r], and/or the [l/ɹ] sound).

All three models show little tendency to misclassify [l] as [ɹ], or vice versa. This indicates that the retraining preserves the distinction between the [l] and [ɹ] phoneme categories.

4.2 Inter-category Distance Ratio For DNN Perceptual Learning

To investigate where the retuning takes place, we examined the effect of increasing amounts of adaptation material on the hidden layers of the models using the inter-category distance ratio proposed in [113]. This measure quantifies the degree to which lexical retuning has modified the feature representations at the hidden layers using a single number. First, the 1024-dimensional vector of hidden layer activations is re-normalized, so that each vector sums to one, and averaged across the frames of each segment. Second, the Euclidean distances between each [l/ɹ] sound and each [l] segment are computed, after which the distances are averaged over all [l/ɹ]-[l] token pairs, resulting in the average [l]-to-[l/ɹ] distance. Third, using the same procedure the average [ɹ]-to-[l/ɹ] distance is computed. The inter-category measure is then the ratio of these two distances and is computed for each of the ten bins.

Figures 5 through 7 show the inter-category distance ratio ([l/ɹ]-to-[l] over [l/ɹ]-to-[ɹ]) for the baseline model, the AmbL model, and the AmbR model, respectively, for each of the 5 hidden layers, for each of the bins.

Figure 5 shows that for earlier bins in the baseline model, the distance between the ambiguous sounds and the natural [l] category and natural [ɹ] category is approximately the same for the different layers, with a slight bias towards [ɹ] (the ratio is >1); the lines for the five layers are close together and do not have a consistent ordering. From bin 5 onwards, and particularly for the last 3 bins, the distance between [l/ɹ] and the natural [l] category decreases from the first (triangles) to the last layer (diamonds), suggesting that [l/ɹ] is represented closer to the [l] category deeper in the neural net. However, this cannot be observed in the classification scores:

Figure 2 shows that [l/ɹ] is primarily classified as [ɹ]. The adaptation of [l/ɹ] towards natural [l] for the later bins suggests that adding training material of the speaker improves the representation of the natural classes as well, because the distance between [l/ɹ] and the natural classes changes without the model being trained on the ambiguous sounds.

Figure 6 shows that, for the AmbL model, the distance between [l/ɹ] and the natural [l] category becomes increasingly smaller deeper into the network: The line showing hidden layer 1 (triangles) is almost always on top, and the line showing layer 5 (diamonds) is almost always at the bottom. Interestingly, there is a downward slope from the first to the last bin, indicating that with increasing numbers of [l/ɹ] training examples labeled as [l], the distance between [l/ɹ] and natural [l] continues to decrease, even though there are no natural [l] tokens in the retuning data. This continual decrease in distance between [l/ɹ] and natural [l] seems to be correlated with the continual increase in classification of the ambiguous sound as [l] for the later bins in Figure 3, and might indicate further adaptation of the representation of the ambiguous sound towards the natural [l].

In the AmbR model (Figure 7), the ratio of distance([l/ɹ],[l]) over distance([l/ɹ],[ɹ]) increases from layer 1 to layer 5, indicating that the neural embedding of [l/ɹ] becomes more [ɹ]-like deeper in the network. So, like the AmbL model, the AmbR model also shows lexical retuning: The speech representation of [l/ɹ] becomes increasingly closer to that of the natural [ɹ] deeper into the model. The effect of increasing amounts of adaptation material is however not as clear-cut as for the AmbL model. The distance ratio rises until bin 2 (8 [l/ɹ] training examples), then falls until bin 5, then rises until bin 7, then falls again. This inconsistency is also found in the classification scores of [l/ɹ] as [ɹ] in Figure 4 but to a lesser extent, which suggests that the increase in the distance between the [l/ɹ] and [ɹ] categories is not large enough to substantially impact classification results.

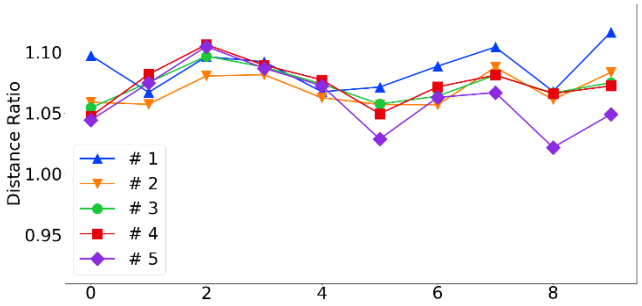


Fig. 5. Ratio of distance([l/ɹ],[l])/distance([l/ɹ],[ɹ]) for the Baseline model.

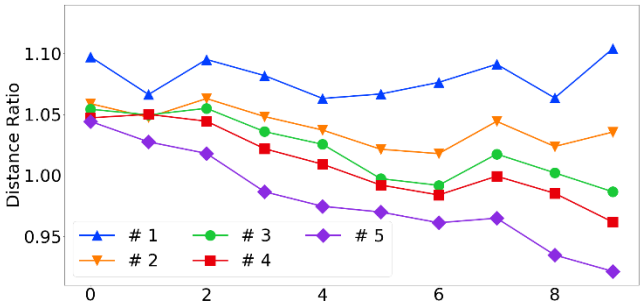


Fig. 6. Ratio of distance([l/ɹ],[l])/distance([l/ɹ],[ɹ]) for the AmbL model.

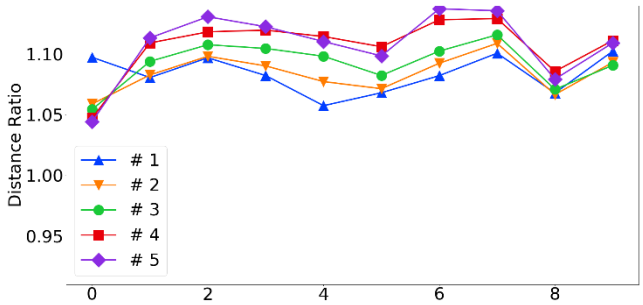


Fig. 7. Ratio of distance([l/ɹ],[l])/distance([l/ɹ],[ɹ]) for the AmbR model.

4.3 Investigating Step-like Behavior For DNN Perceptual Learning

As the classification rates make a significant jump after just seeing the first bin of words for all three experimental sets, which indicates very fast adaptation, in the second experiment, we investigate how the CGN-only model adapts to a single bin of retuning data over the training course in the very first time step. Similar to the procedure above, we evaluate the classification rates by training the CGN-only model using the first training bin (training bin 0) from each experiment set (natural, AmbL, AmbR) for 30 epochs. Before the first epoch of training (t=0), and after each epoch of training (1≤t≤30), we record the percentage of [l], [ɹ], and ambiguous [l/ɹ] sounds from the second test bin (test bin 1) that are classified as either [l] or [ɹ] (a total of 31 time points, 0≤t≤30).

Figure 8 shows the classification rates over 30 epochs for the natural model using natural stimuli from the first training bin: both [l] and [ɹ] sounds show immediate adaptation after the first epoch (correct response rate increases by about 20% from t=0 to t=1). The [ɹ] sound shows the highest accuracy over 30 epochs, but the number of [ɹ]'s correctly recognized only increases very slightly after the fifth epoch. After reaching a peak by the first epoch, the classification rate for [l] decreases until the third epoch, and then flatlines (with some small oscillations). Interestingly, while ambiguous [l/ɹ] sounds are not present in the training data, more and more

[l/ɹ] get classified as [ɹ] as training progresses, meaning that the bias of [l/ɹ] toward [ɹ] somehow increases without the model seeing any ambiguous sounds.

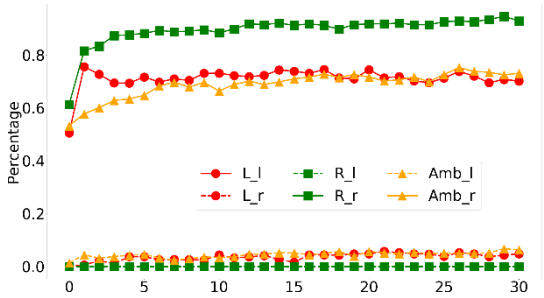


Fig. 8. Proportion of [l] and [ɹ] responses by the natural model over 30 epochs for the first bin.

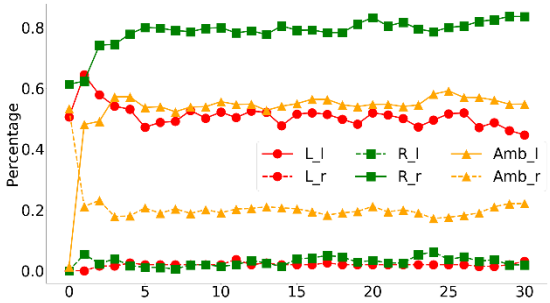


Fig. 9. Proportion of [l] and [ɹ] responses by the AmbL model over 30 epochs for the first bin.

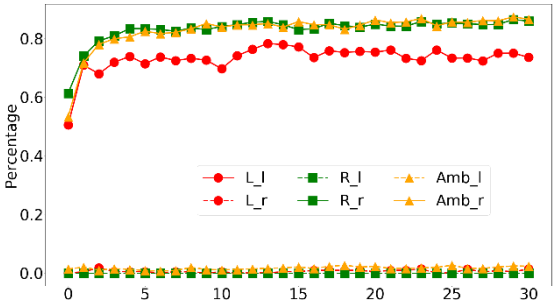


Fig. 10. Proportion of [l] and [ɹ] responses by the AmbR model over 30 epochs for the first bin

Figure 9 shows the classification rates over 30 epochs for the AmbL model using stimuli from the first training bin with ambiguous sounds labeled as [l]. The classification rates at t=0 are the same in Figures 8 and 9 because they are based on the same model; it is only after the first training epoch (t=1) that their rates diverge. Similar to Figure 8, the accuracy for [ɹ] reaches 80% within 5 epochs, with a large jump at the second epoch. The accuracy for natural [l] also jumps

up after the first epoch, even though there are no [l] tokens in the training data, but beginning with the second epoch, the model starts to forget how to correctly classify natural [l] tokens. The most important observation comes with the ambiguous [l/ɹ] sound. After just a single epoch on a single bin of data, the percentage of [l/ɹ] sounds classified as [l] goes from 0% to a little below 50%. However, after 5 epochs, the accuracy for [l/ɹ] as [l] flatlines around 50%, meaning that the model has reached its limit of perceptual learning by seeing only one training bin.

Figure 10 shows the classification rates over 30 epochs for the AmbR model using stimuli from the first training bin with ambiguous sounds labeled as [ɹ]. While no natural [ɹ] is present in this experiment set, the accuracy for natural [ɹ] gradually increases until the fifth epoch, meaning that perceptual learning on ambiguous sounds as [ɹ] also helps the model learn a natural [ɹ]. The ambiguous [l/ɹ] sound is classified as [ɹ] 50% of the time at t=0, i.e., with no training; the t=0 case is identical to those shown in Figures 8 and 9. After just one epoch of training, using one bin of ambiguous sounds labeled as [ɹ], the model learns to perform this classification with 70% accuracy, and accuracy increases until the fifth epoch.

It is worthwhile, at this point, to remind the reader what is meant by “one epoch” in the training of a neural net. Each epoch of training consists of three stages: (1) a direction vector *d* is chosen; in the first epoch, this is just the negative gradient of the error; (2) a search procedure is used to choose the scale, *g*; (3) the neural network weights are updated as $w=w+gd$. Each epoch of training can only perform a constant shift of the previous network weights. Figures 2-4 and 8-10 show that most of the DNN adaptation occurs in the first epoch on the first bin of the adaptation material, i.e., on the first update of the direction, therefore most of the DNN adaptation can be characterized as a constant shift in the network weights. This makes sense since the model is just learning about 4 additional training tokens (one adaptation bin) — with only 4 tokens, while it is not possible to learn a very complicated modification of the boundary, learning a boundary shift is indeed possible and very likely the case here.

In a deep neural network, a constant shift of the network weights is not the same thing as a constant shift of the classification boundary, but in practice, the revision of w after the first epoch is usually not much more complicated than a shifted boundary. The finding that inter-talker adaptation can be accomplished by a constant shift in cepstral space is not new; it has previously been reported by [101]. The finding that a comparable constant shift is sufficient to learn distorted sounds, like the ambiguous [l/ɹ] sound, has never previously been reported.

4.4 Visualizing Phoneme Boundary Shift For DNN Perceptual Learning

To visualize how the phonetic boundaries shift for the Baseline, AmbL and AmbR models, the activation vectors for the phonetic segments, as defined by forced alignment, were combined together as a single utterance matrix and fed into the NFA algorithm. Note that in order to visualize the time course, the summary vectors were jointly extracted for all utterance matrices across the 9-step adaptation course. As mentioned earlier, NFA was trained on phonetic segments from all phone classes (plus the extra ambiguous sound class). Following that, PCA was trained on only the summary vectors from the natural /l/, natural /r/ and ambiguous [l/r] class. The results for the Baseline, AmbL and AmbR are plotted in Tables 1–3.

Table 1. Visualization for the time-course of adaptation for the Baseline DNN model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

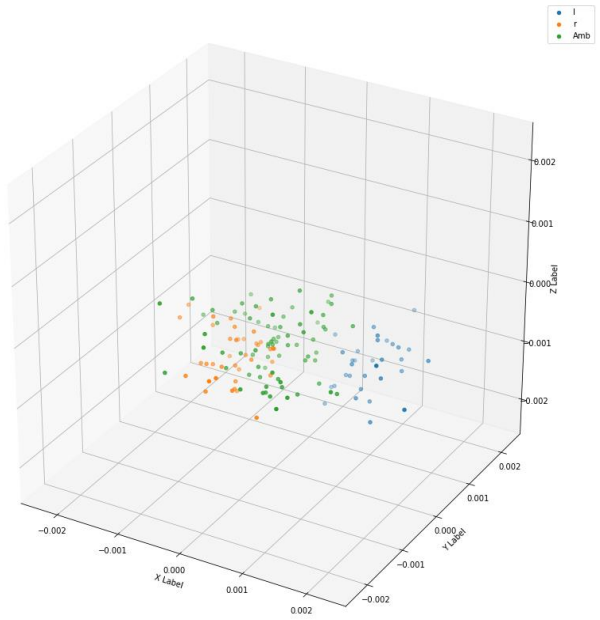
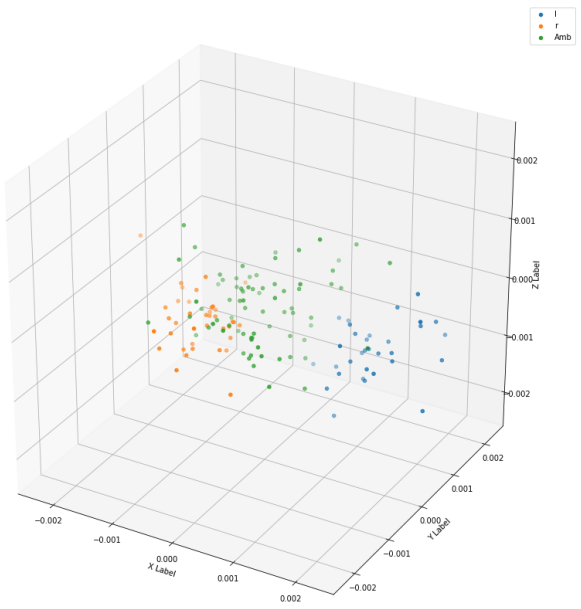
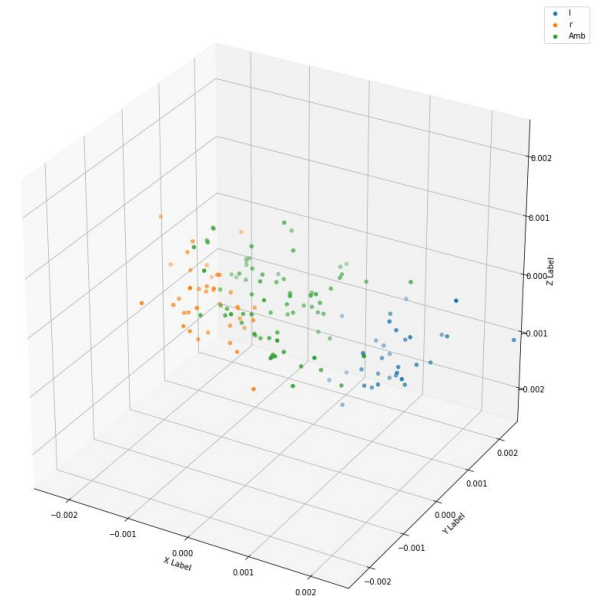
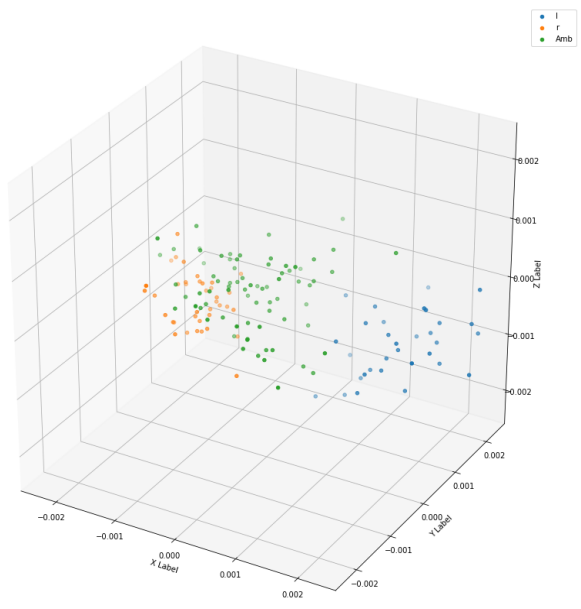
Step 0: 	Step 1: 
Step 2: 	Step 3: 

Table 1 Continued

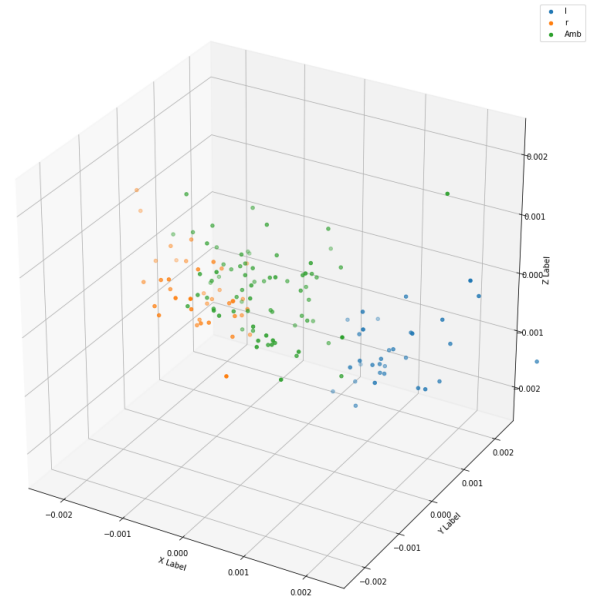
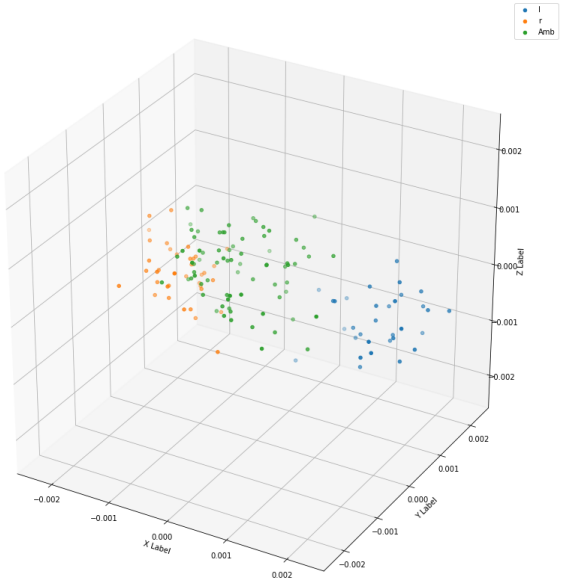
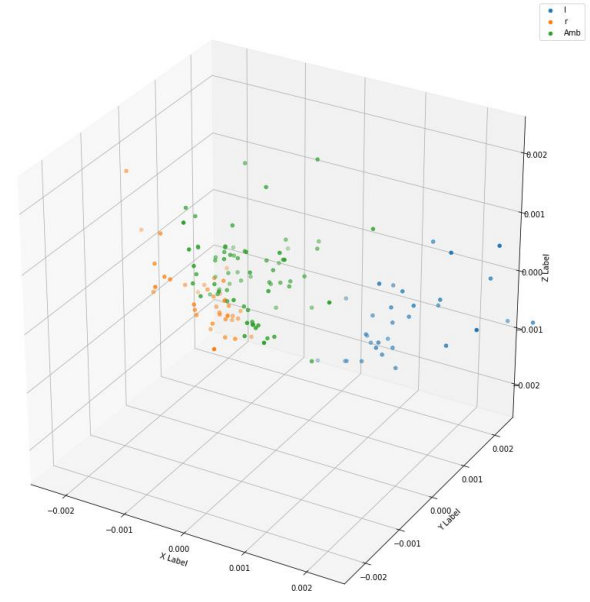
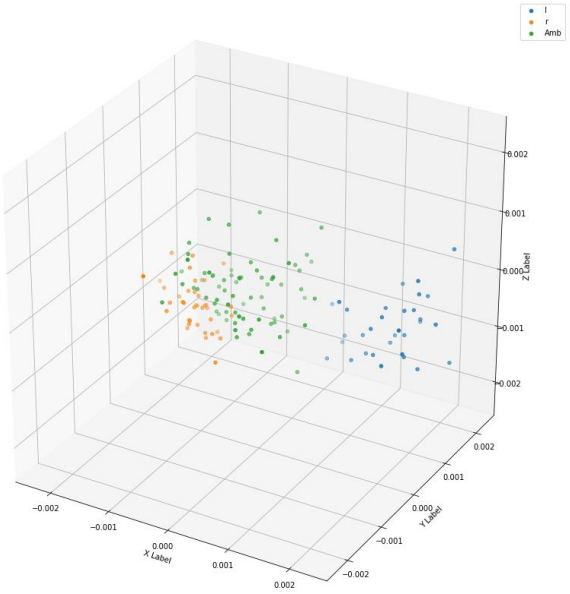
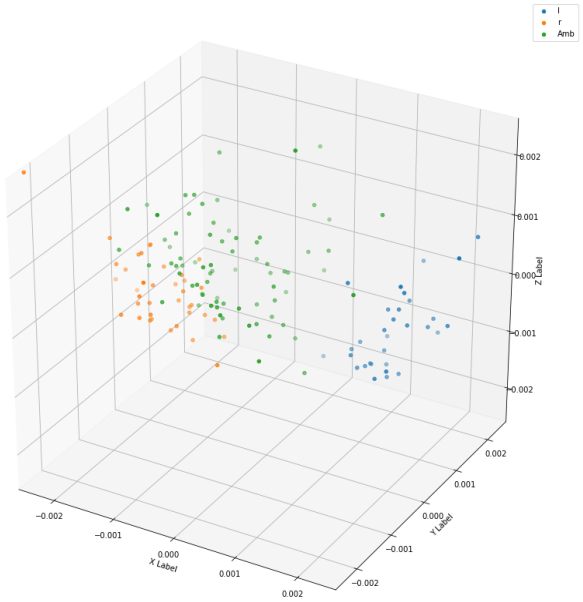
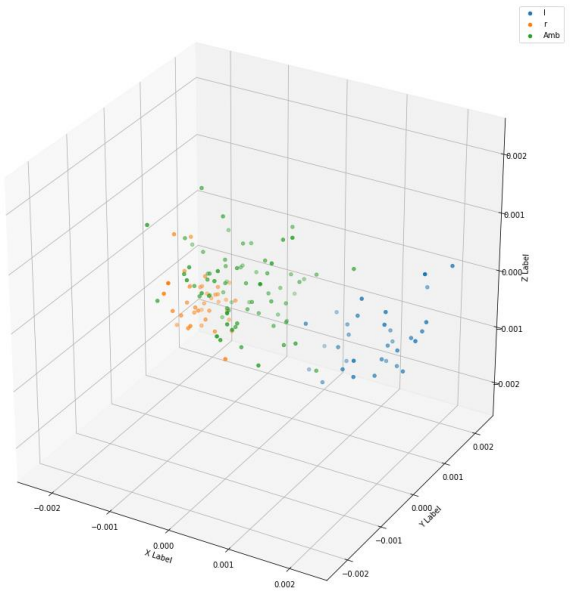
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 1 Continued

Step 8:	Step 9:
	

For the Baseline model shown in Table 1, the ambiguous sounds stand somewhere in the middle of natural /l/ and /r/ before retuning happened (step 0), but the bias towards /r/ increases as subsequently more training bins are fed into the network. The bias towards /r/ seems roughly fixed after step 2, although subsequent steps of re-tuning generate tighter clusters at some time step (such as steps 5 and 7). This agrees with our classification results established earlier on the Baseline model (Figure 2), as the classification rates of ambiguous sound also show a bias towards natural /r/, and that the classification rates also remain relatively constant after step 2. Likewise, in the visualization shown above, the clustering behavior after step 2 does not show too much of a change boundary-wise, and behaviors such as slight cluster shrinks and expansions are likely due to the result of weight fine-tuning when the DNN fails to learn anything new to improve its performance.

Table 2. Visualization for the time-course of adaptation for the AmbL DNN model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

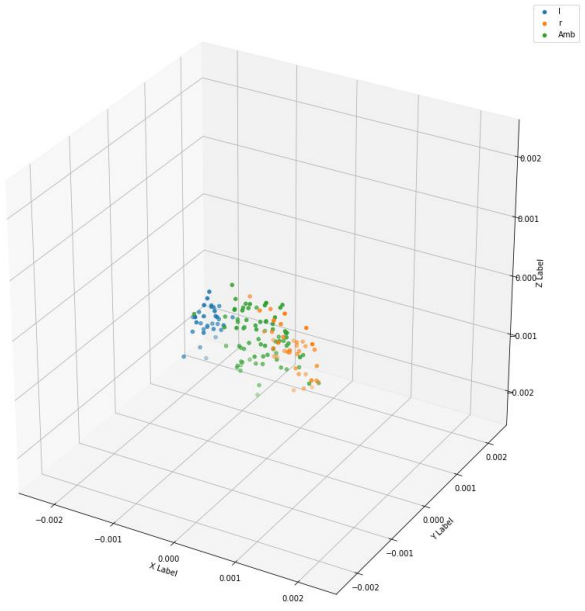
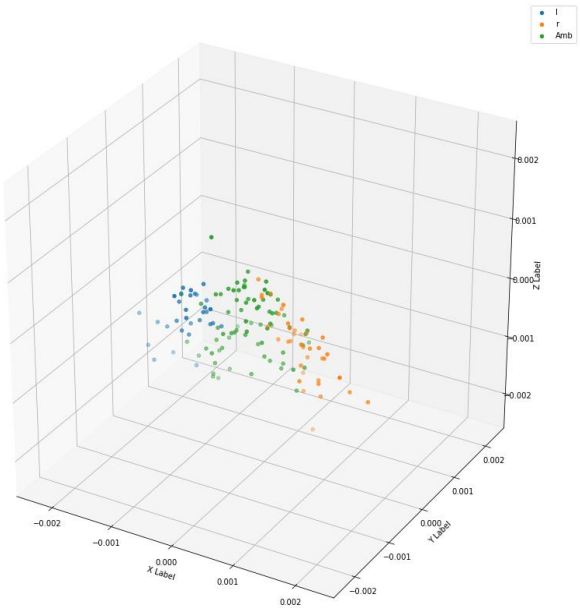
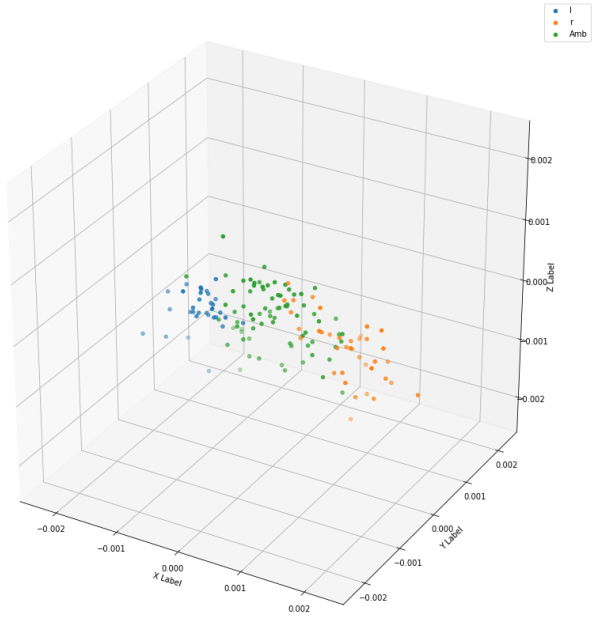
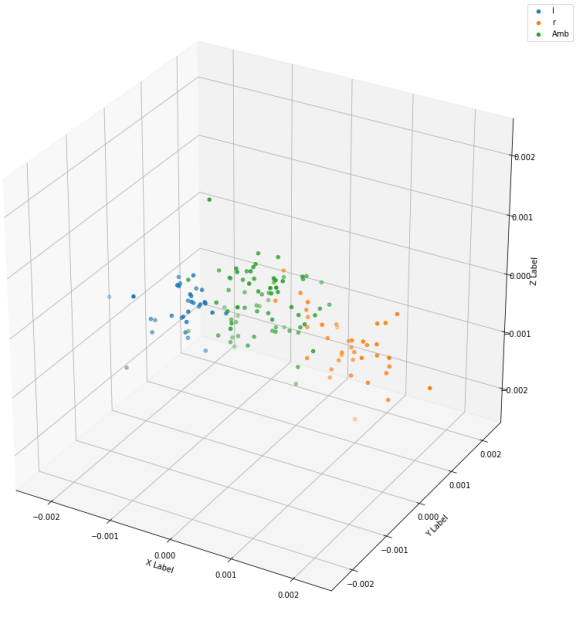
Step 0: 	Step 1: 
Step 2: 	Step 3: 

Table 2 Continued

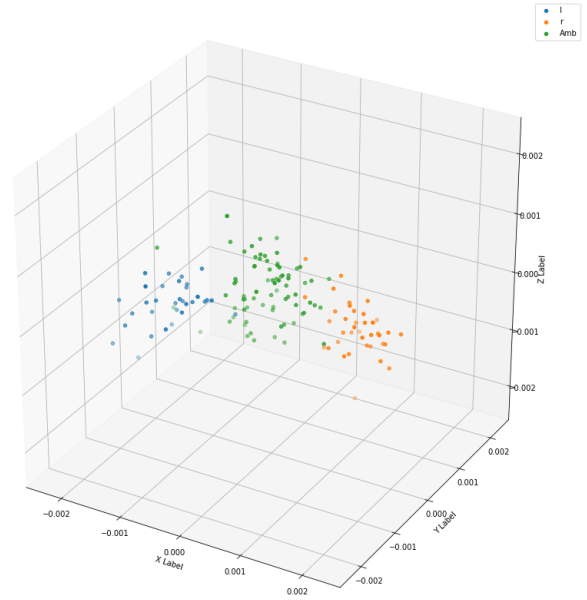
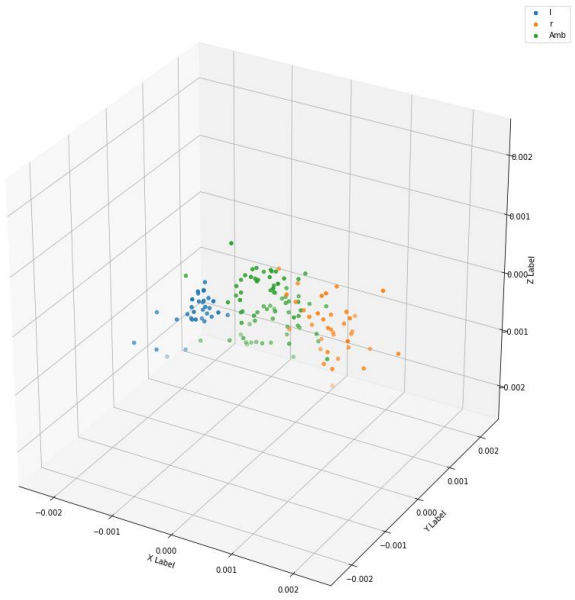
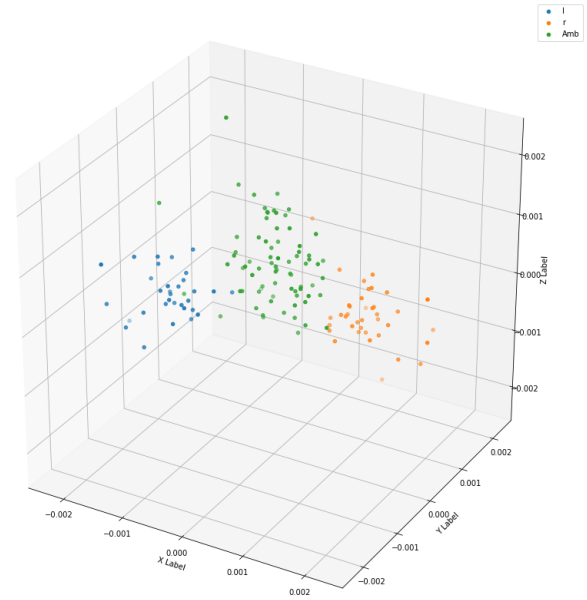
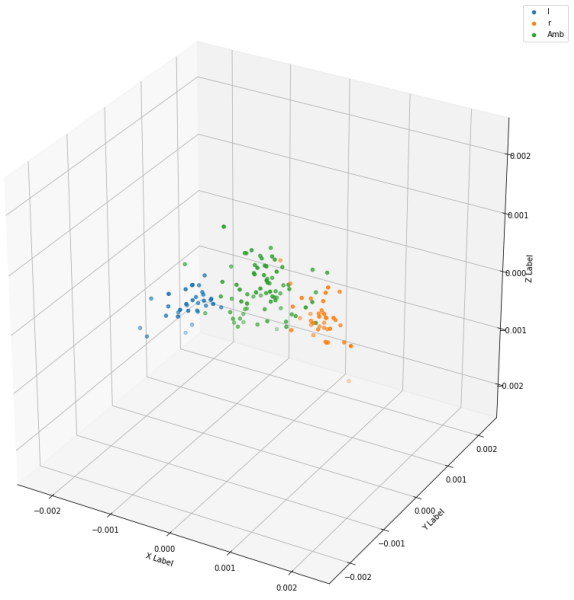
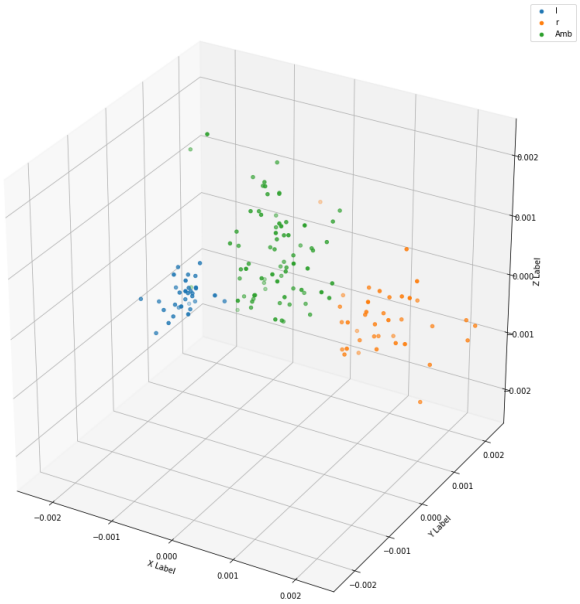
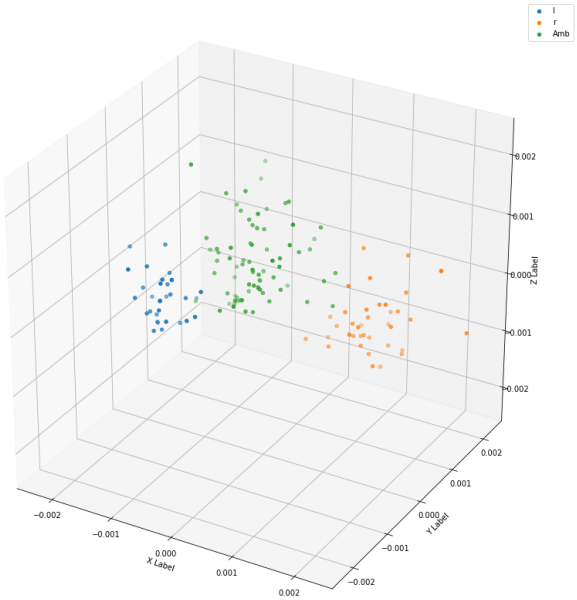
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 2 Continued

Step 8:	Step 9:
	

The results from the AmbL model shown in Table 2, however, can be a bit harder to interpret. Across all time steps, the ambiguous sounds just stand somewhere in the middle of natural /l/ and /r/, and subsequent steps of adaptation merely increase the spacing between the three clusters. However, compared with the Baseline model, some effort of perceptual learning could still be observed, as the ambiguous sounds get more and more separated from their initial bias toward natural /r/ sound. The effort is more gradual than step-like in this case, and this is analogous to the gradually rising Amb_l curve shown in Figure 3.

However, the model fails to collapse the [l/r] cluster with the natural /l/ cluster, which is a little unexpected as the course of adaptation should show a shift towards the /l/ cluster after perceptual learning (i.e. labeling ambiguous sounds as /l/ during training). The reason could be due to the interesting phenomenon that the model “forgets” what a natural /l/ should sound like during perceptual learning (note that from Figure 3, the percentage of natural /l/ classified as /l/ was below 40% after step 4 and continues to worsen). It is likely that, just as shown above in visualization, the natural /l/ cluster has shifted away from its canonical representation.

Table 3. Visualization for the time-course of adaptation for the AmbR DNN model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

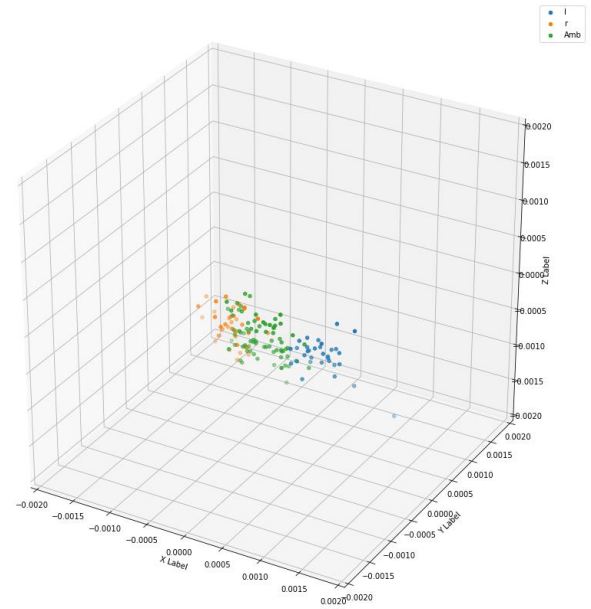
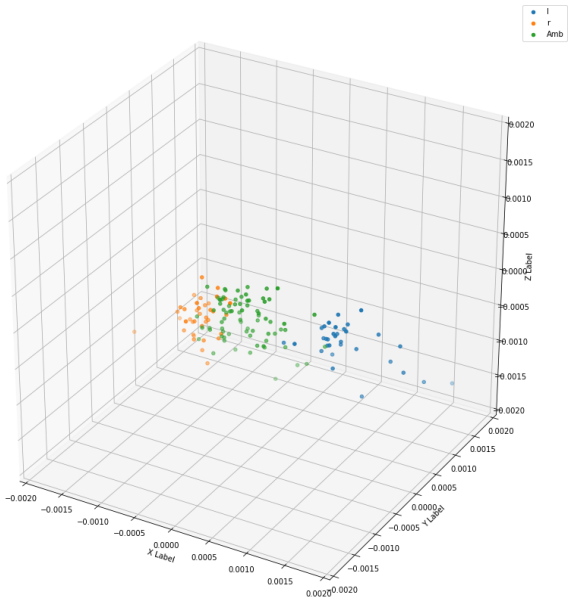
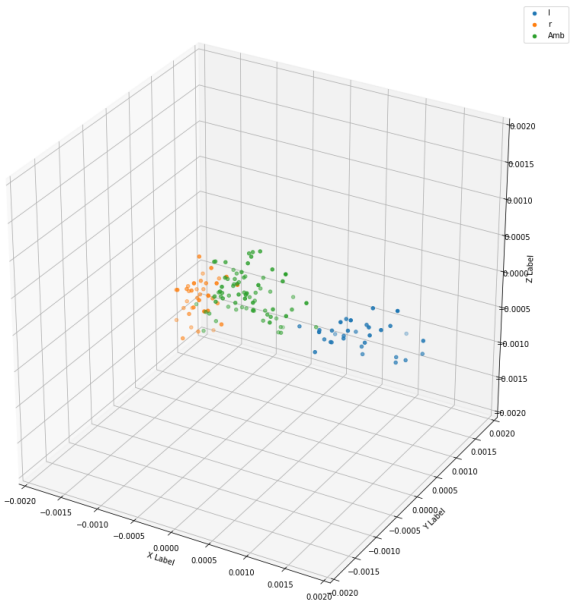
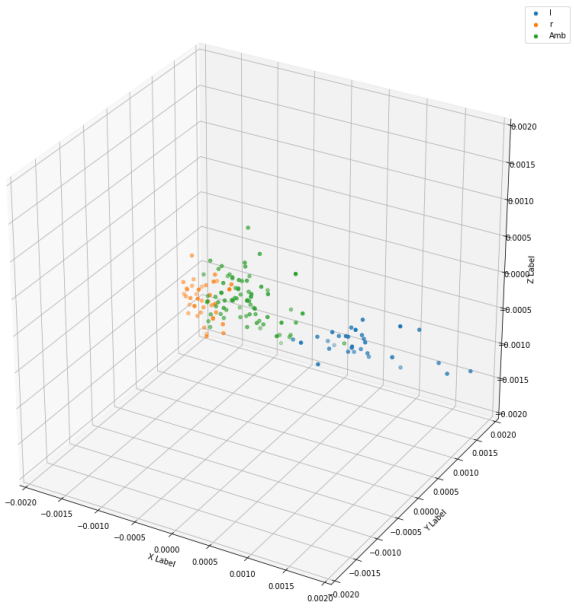
Step 0: 	Step 1: 
Step 2: 	Step 3: 

Table 3 Continued

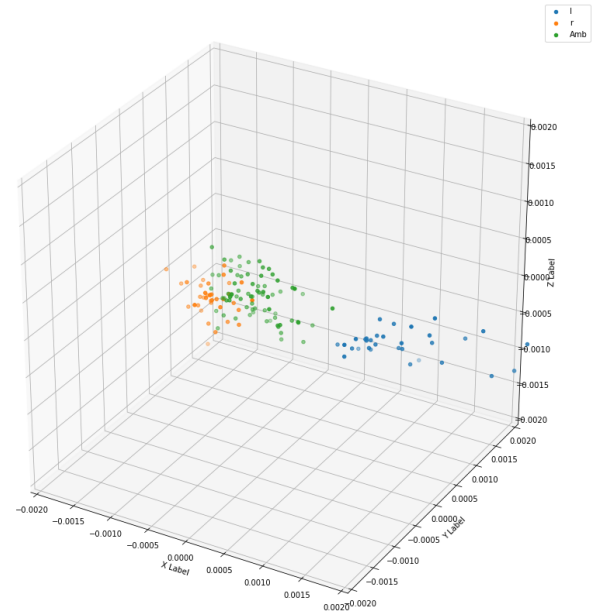
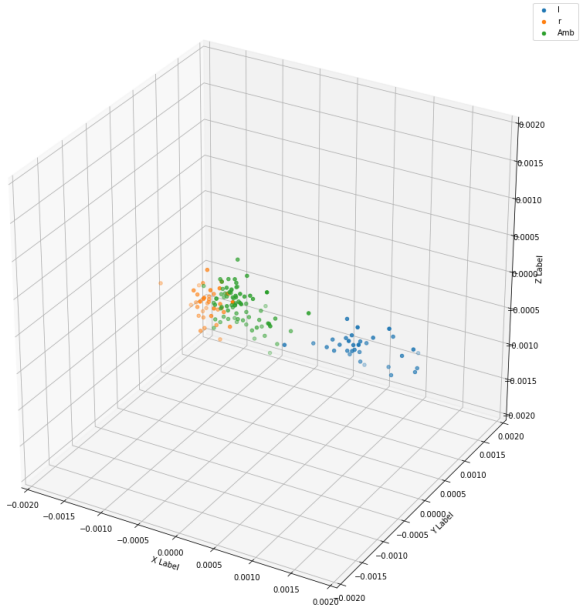
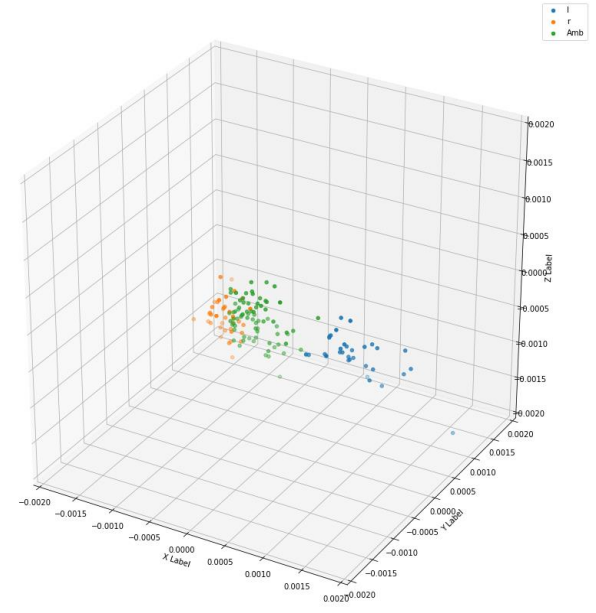
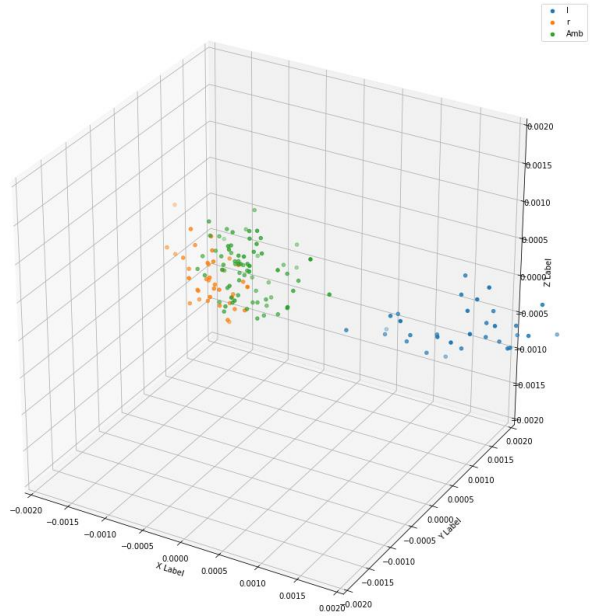
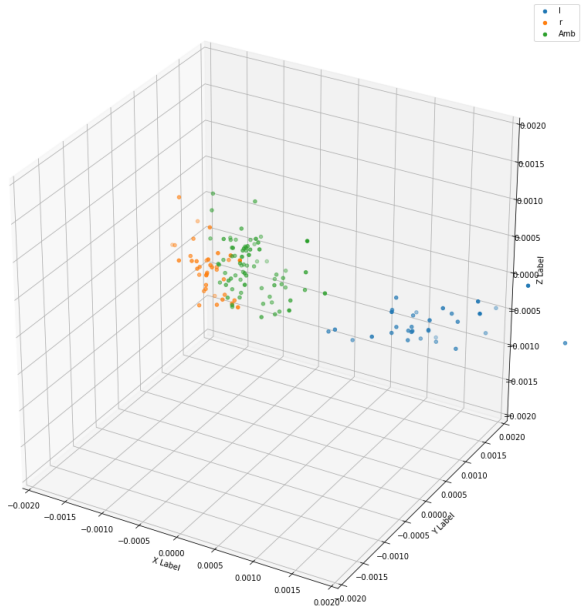
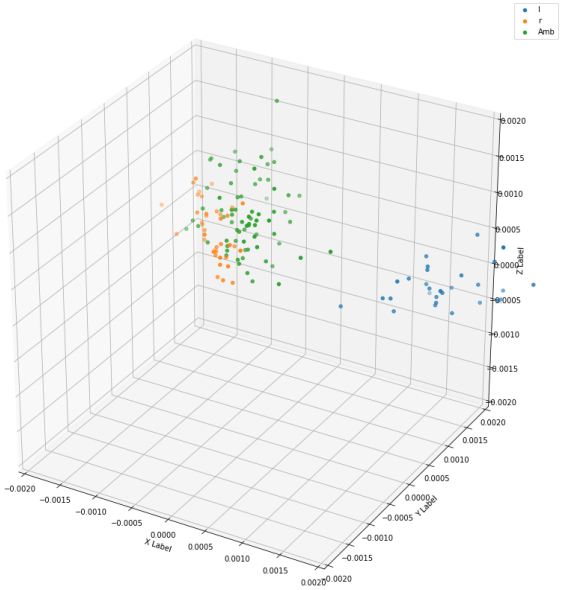
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 3 Continued

Step 8:	Step 9:
	

The visualizations on the AmbR model across the time-course shown in Table 3 are more interpretable. At time step 0 before adaptation takes place, the ambiguous sounds lie somewhere in the middle of natural /l/ and /r/, and the three clusters were relatively close to each other. Starting from step 1, however, an immediate perceptual learning effect could be observed: the ambiguous sounds “joined” the cluster for natural /r/ sounds, with a well-separated boundary from the natural /l/ sounds. This agrees with the previous results on the classification rates of the AmbR model across time-course (Figure 4), as the perceptual learning effect is also step-like according to the Amb_r curve, after just seeing the first training bin.

There are two interesting effects worth mentioning: (1) In the visualization it is easy to see that the model merged the clusters for ambiguous sound and natural /r/, instead of a naive decision boundary shift. This indicates that perceptual learning happens through changing the hidden representation of the ambiguous sounds. (2) While the classification rates in Figure 4 remained somewhat constant after step 1 or 2, changes in clustering behaviors are still evident in later bins for the visualizations shown above, and the merged cluster of ambiguous sounds and natural /r/ gets further pushed away from natural /l/. This means that although the perceptual learning effect is step-like, increasing the number of adaptation tokens still helps the DNN learn a better internal representation. The better representation does not have an effect on classification rates, however, as the boundary is already clear-cut after step 1.

4.5 Recognition Rate for Bi-LSTM Perceptual Learning

In this experiment, we repeated what we did for our very first experiment to determine the number of tokens needed to start observing the perceptual learning effect during the adaptation course of the Bi-LSTM ASR model. The recognition rate was calculated by simply counting the percentage of the word-final /l/ (or /r/, or ambiguous [l/r]) recognized as phoneme /l/ or /r/ that occurred at the end of the generated transcripts. Figure 11 through 13 show the results, with solid lines at the top for the correct label, and dashed line at the bottom showing the incorrect label (for [l/r] it follows the label it saw during adaptation). The meanings for the different labels are the same as in 4.1

Figure 11 shows the results for the Bi-LSTM baseline model retrained with the natural stimuli and their transcriptions, under CTC criterion. The [r] sound starts a little below 80% averaged (which is higher than that of DNN prior to any retuning) and after seeing one bin, rises to >90%. The [l] sound also starts a little below 80% (which is also higher than that of DNN), rises to >90% by step 3, drops to 80% again, and slowly rises again to > 90%. The [l/ɹ] sound (not part of the training data for this model) is recognized as [ɹ] about 80% of the time, which shows an even larger bias than the naive DNN, and as [l] about 10%- 20% of the time.

Figure 12 shows the results for the Bi-LSTM model retrained with the ambiguous sound labeled as [l] in the transcriptions, under CTC criterion. The recognition rate for natural /r/ first drops a bit and then remains >90% after step 3. The recognition rate for natural /l/, interestingly, remains below 60% after step 2, and while somewhat noisy across the time course, clearly showed a decreasing trend. This is very similar to our results in 4.1, as the DNN also seemed to have forgotten what a natural /l/ sounded like after the adaptation.

The most important behavior in Figure 12 is the step-like increase of recognition rate for [l/r]: the rate is almost zero prior to any training, but after seeing only 4 words with their word-final ambiguous sound labeled as /l/ in the phonetic transcription, the rate quickly rises to 80%. This again indicates perceptual learning for the AmbL model, and as discussed earlier, is very similar to the step-like response from a human listener [27, 95, 102, 109, 112, 114].

Subsequent adaptation on the ambiguous sound first shows a slight dip and rises to around 90% by step 4, and then flatlines.

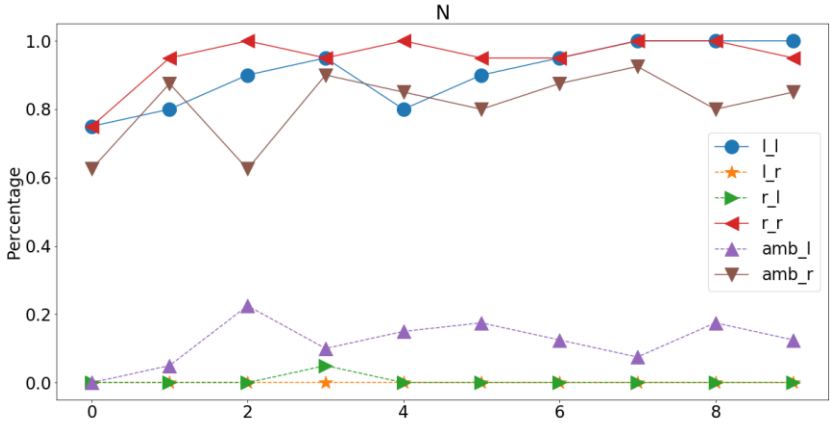


Fig. 11. Proportion of [l] and [ɹ] responses by the baseline model, retrained with natural stimuli, per bin (Bi-LSTM model)

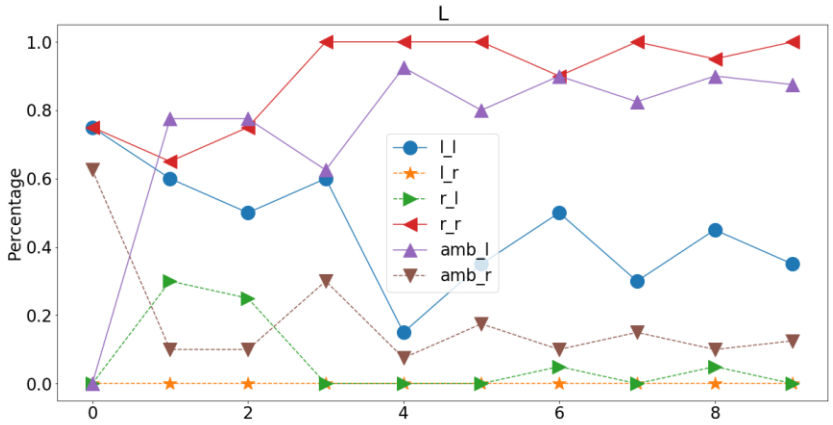


Fig. 12. Proportion of [l] and [ɹ] responses by the AmbL model, retrained with [l/ɹ] labeled as [l], per bin (Bi-LSTM model)

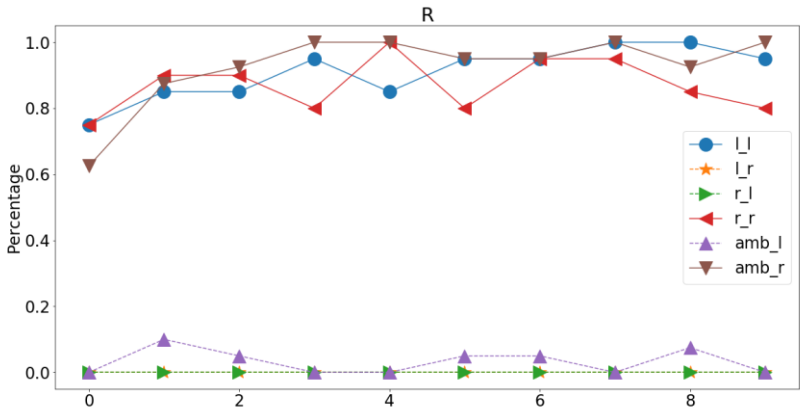


Fig. 13. Proportion of [l] and [ɹ] responses by the AmbR model, retrained with [l/ɹ] labeled as [ɹ], per bin (Bi-LSTM).

Figure 13 shows the results for the Bi-LSTM model retrained with the ambiguous sound labeled as /r/ in the transcriptions, under the CTC criterion. As in 4.1, unlike the AmbL model, the AmbR model has not forgotten what a natural /r/ sounds like, although the recognition rates are around 80% across the time course, which are around 10% lower than those of the baseline natural model, and also slightly lower and noisier than the same curve in Figure 4. More importantly, however, within merely a single time step, the AmbR model also successfully improves its recognition rate for ambiguous sounds (as /r/) from a little over 60% to around 85%, thus showing perceptual learning. Subsequent adaptation steps further improve the recognition rate of [l/r] as [r] up to nearly >95% and flatlines after step 4.

It is important to note that while the model architecture, training criterion, input/output format, and even the method used to calculate recognition rate, differ significantly from those of the Bi-LSTM model to the DNN model in 4.1, the three plots across the time course share significantly more similarities than differences, especially in terms of how many tokens are need for perceptual learning to occur. The results in this section further validate the conclusions and observations made in section 4.1. As mentioned in Chapters 1 and 3, however, it is believed that the Bi-LSTM model better resembles the case of lexically-guided perceptual learning, as the final decision of the Bi-LSTM model between outputting an /l/ or /r/ takes the context of the entire word into account.

4.6 Visualizing Phoneme Boundary Shift For Bi-LSTM Perceptual Learning

To gain more information about if/how the decision boundaries shift during the course of adaptation for Baseline, AmbL and AmbR model, the summary vectors for each phonetic segment as defined by the implicit alignment of CTC output were first jointly extracted for all

phone classes, with those corresponding to natural /l/, natural /r/ and ambiguous [l/r] later fed into PCA for plotting onto the first three principal axes. The plots for all three retuning sets are shown in Tables 4–6.

Table 4. Visualization for the time-course of adaptation for the Baseline Bi-LSTM model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

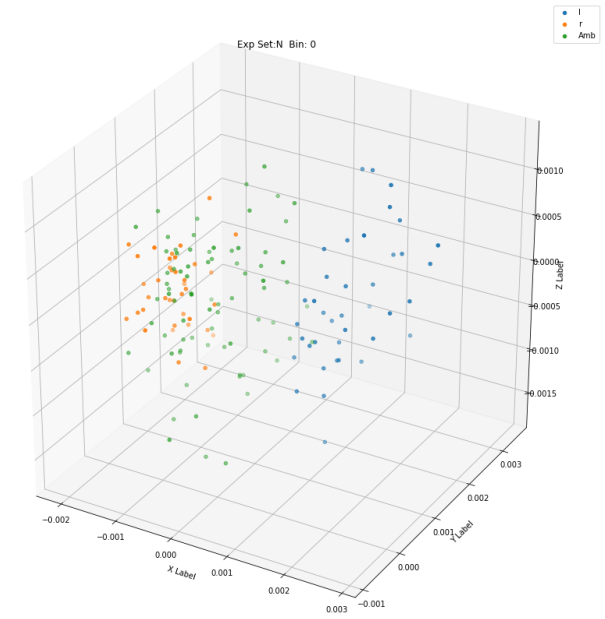
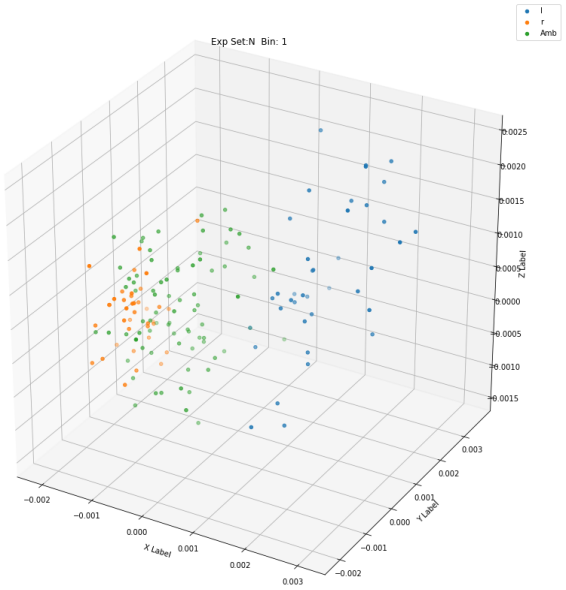
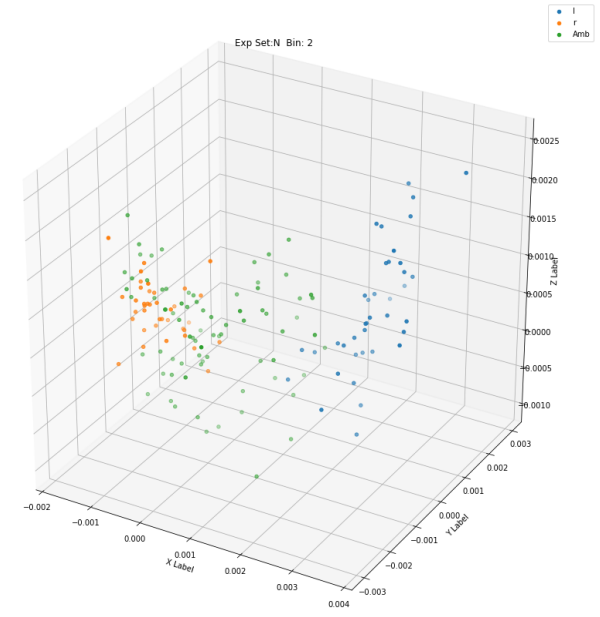
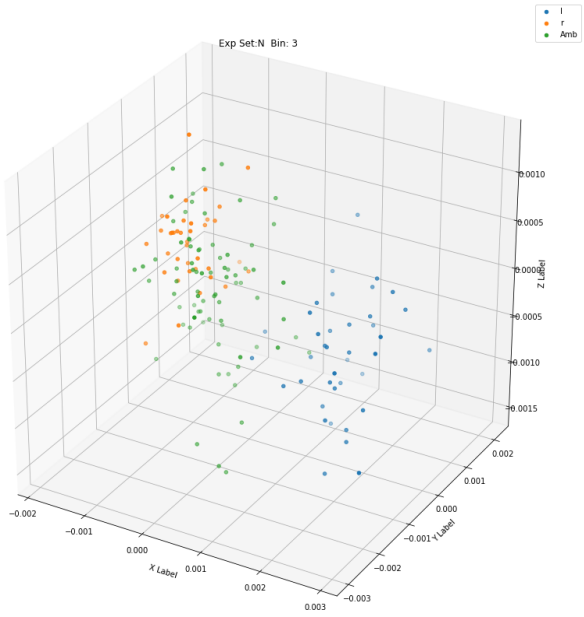
Step 0: 	Step 1: 
Step 2: 	Step 3: 

Table 4 Continued

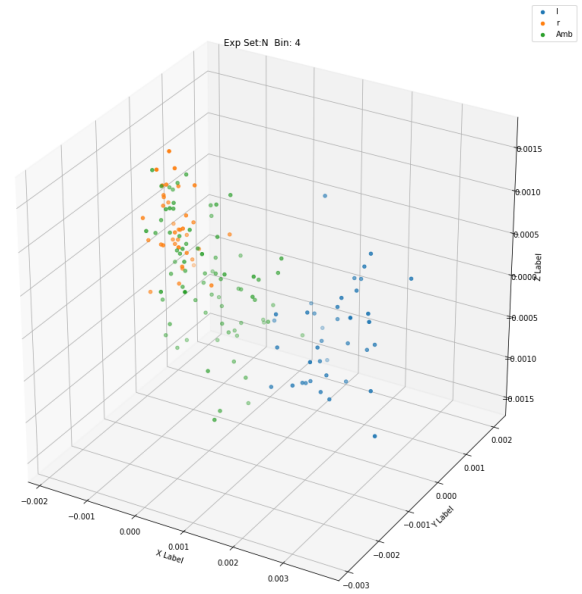
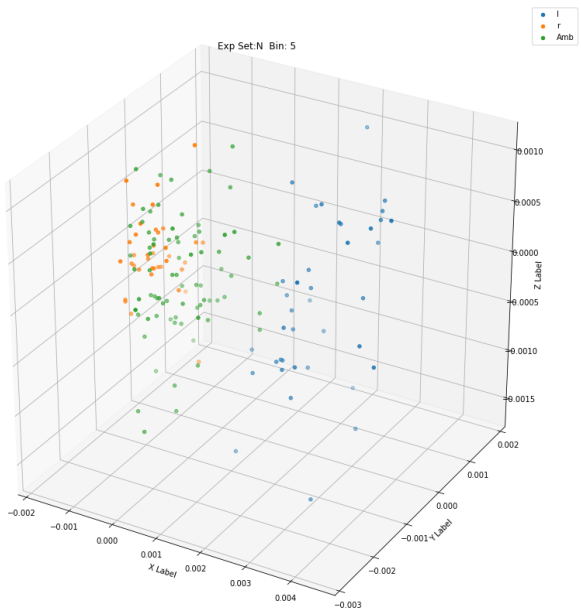
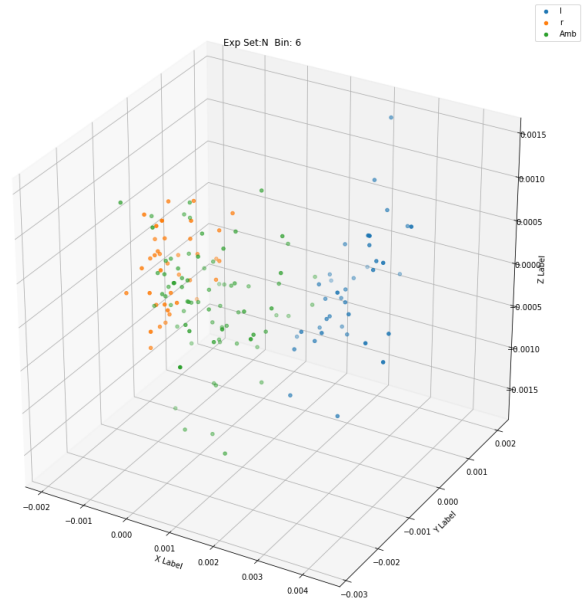
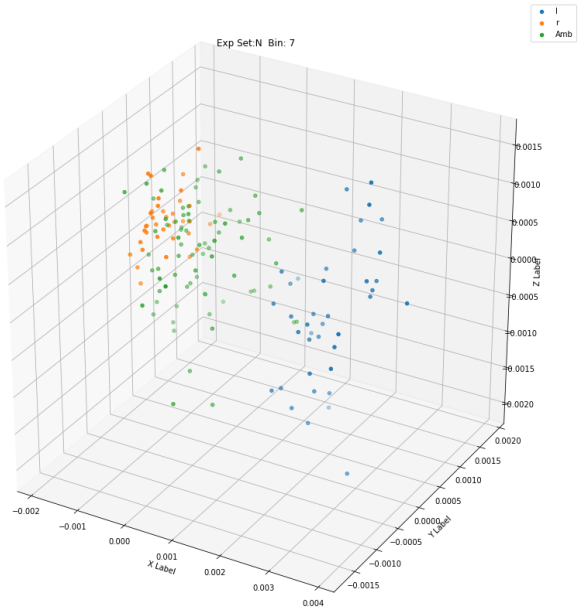
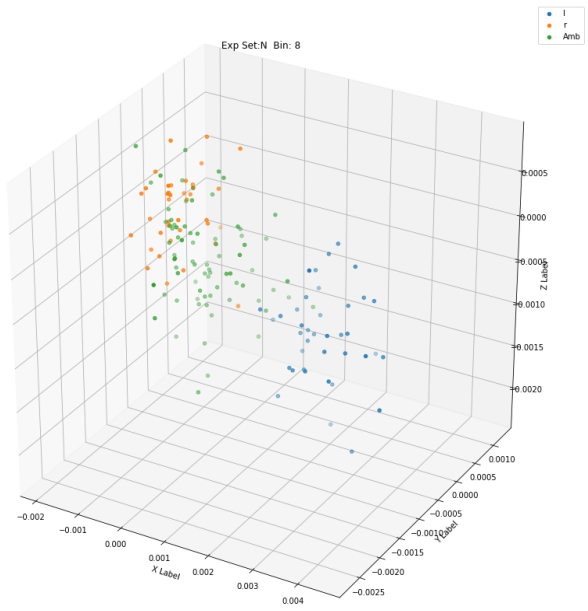
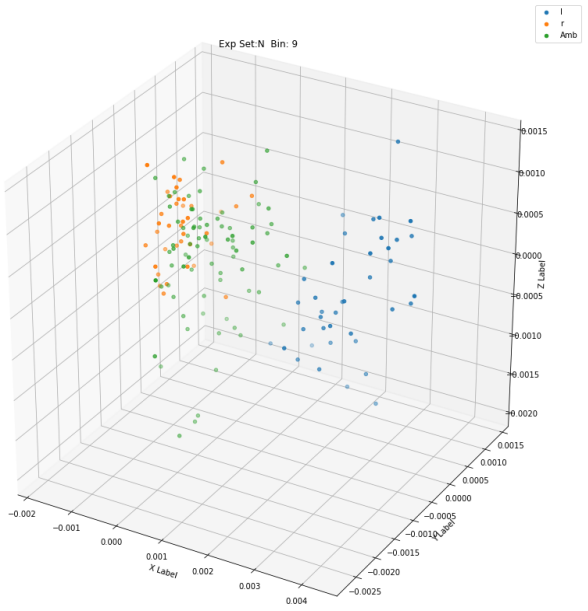
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 4 Continued

Step 8:	Step 9:
	

For the Baseline model shown in Table 4, the bias of the ambiguous sounds to natural /r/ is already pretty obvious before re-tuning happened, and further increases after the first time step as the model sees more natural stimuli, even though none of them actually contain any ambiguous sound. This agrees with the recognition rates in section 4.5, as more than 60% of the ambiguous sounds are recognized as /r/ at step 0, and rises to >85% at step 1.

It is interesting to note that, although the recognition rates for /r/ and /l/ in section 4.5 are comparable (or even higher) than those of section 4.1, the clusters for /r/ and /l/ displayed here are not as tight as those in section 4.4. The reason for this may be two-fold: first, a hidden layer of a bidirectional LSTM may contain not only information about the output phone of at the current time frame, but also other contextual information around it, and therefore the clusters could be more spread out as the contextual information around the same phone in different words is likely to be different; second, while it is viable to use the implicit alignment of the CTC decoder for extracting phonetic segments, the temporal accuracy might not be as good as those from a forced alignment by an HMM model, and thus may also cause the slightly more spread-out clusters observed here.

Table 5. Visualization for the time-course of adaptation for the AmbL Bi-LSTM model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

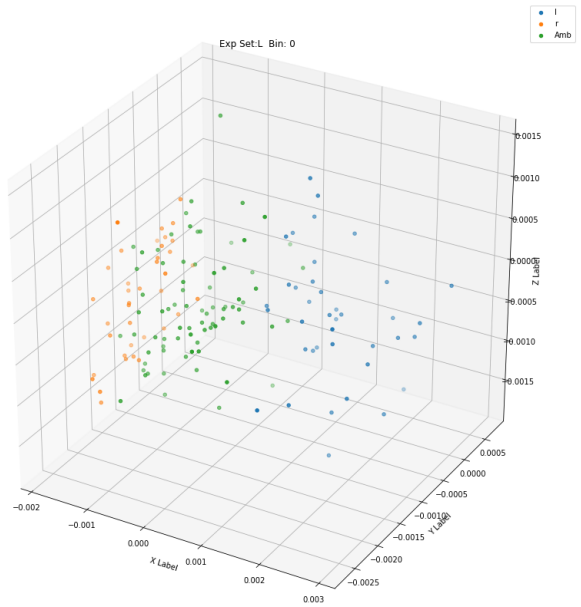
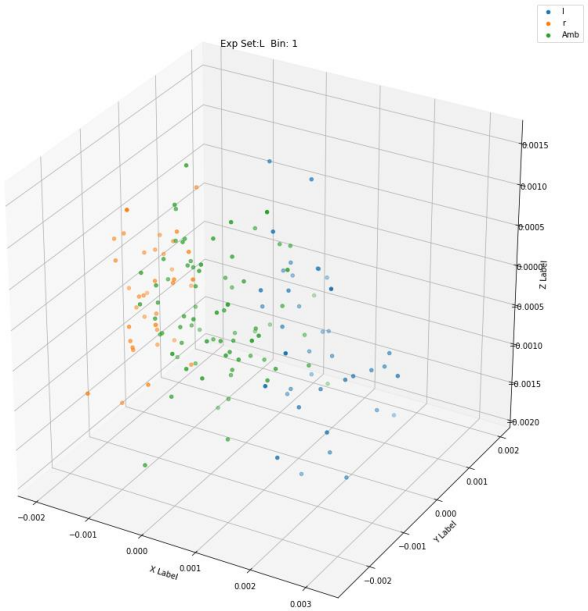
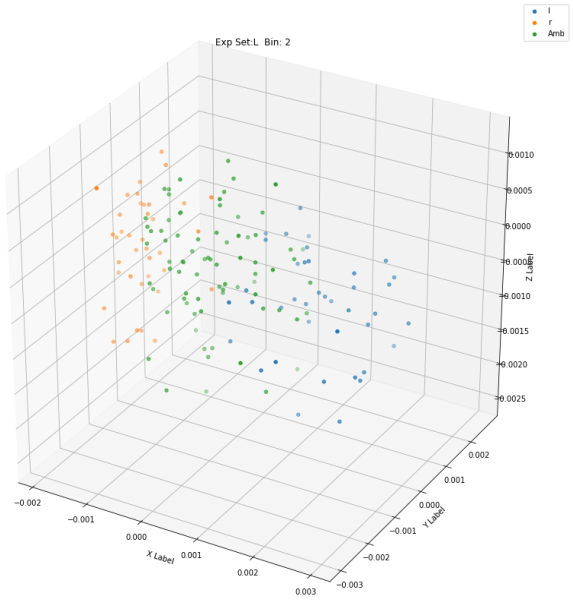
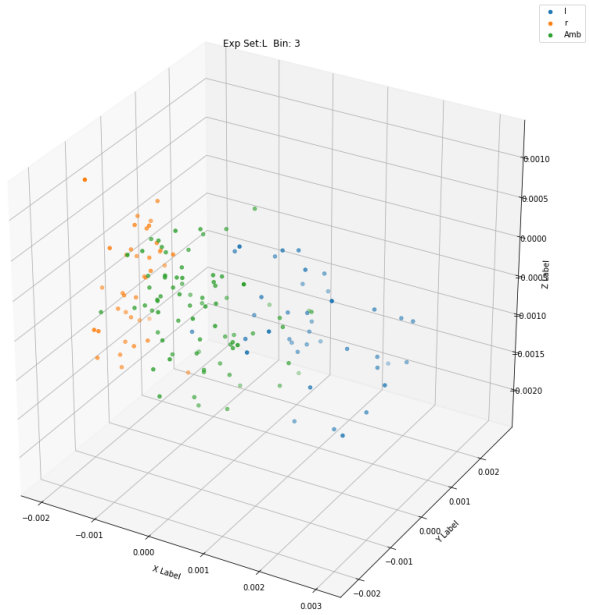
Step 0: 	Step 1: 
Step 2: 	Step 3: 

Table 5 Continued

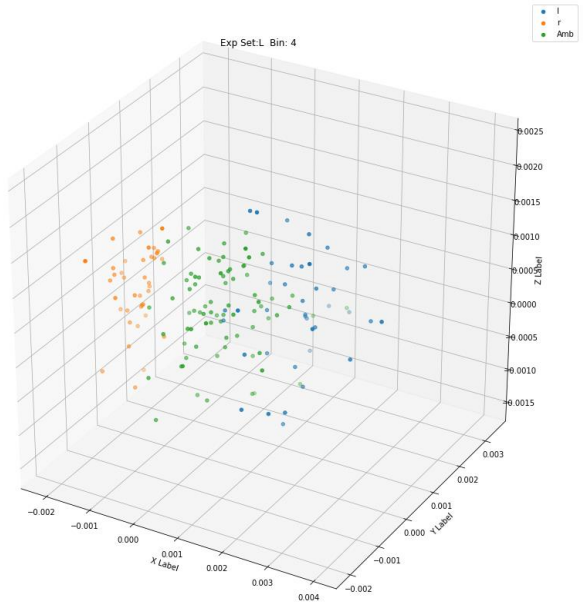
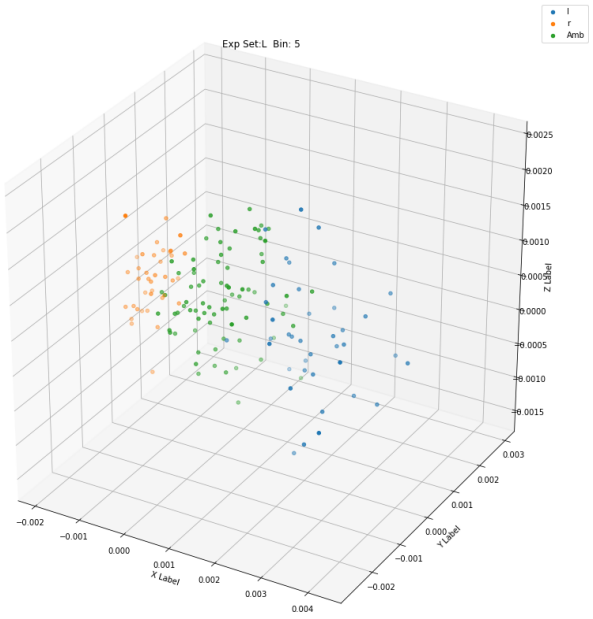
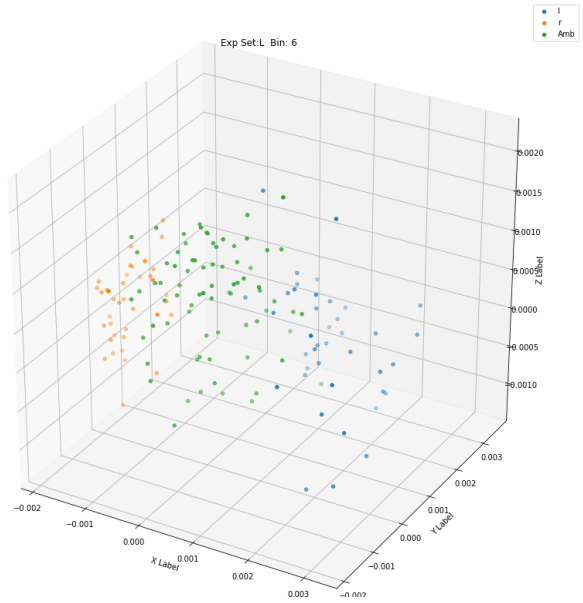
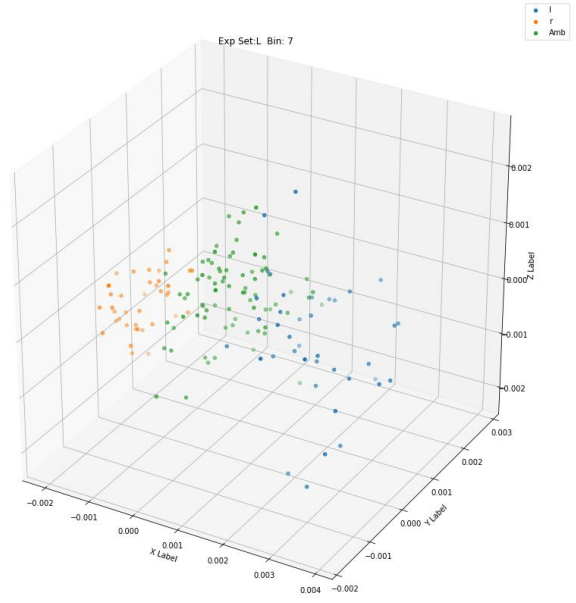
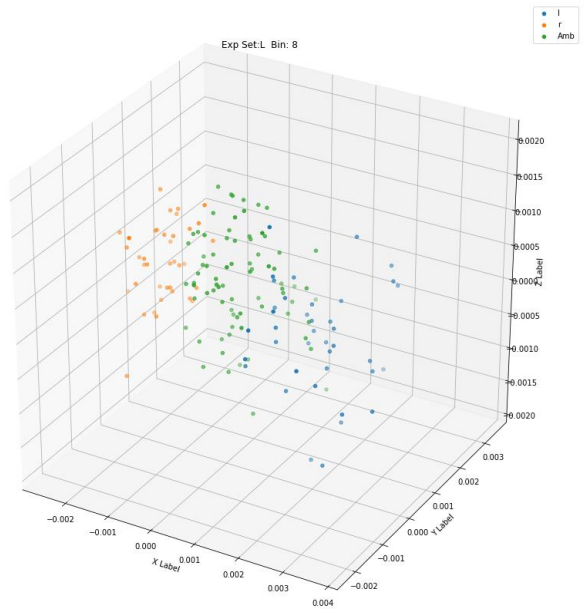
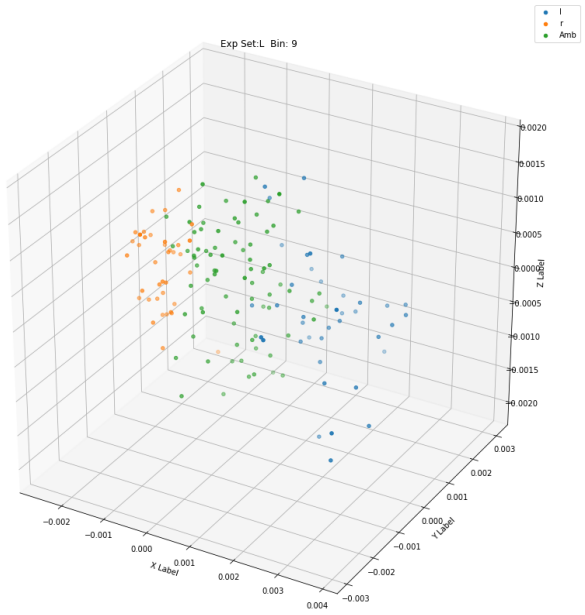
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 5 Continued

Step 8:	Step 9:
	

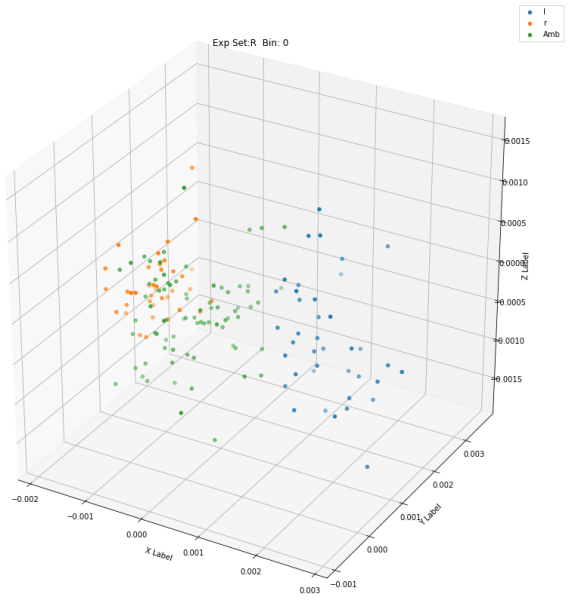
Like the visualization results in Table 2, the ambiguous [l/r] cluster in Table 5 also does not merge with /l/ cluster. This agrees with the recognition rates in Figure 12 for natural /l/, as the AmbL model seems to have forgotten what a natural /l/ sounds like. One speculation would be that during adaptation, it may have changed its decision boundaries for /l/ by including the ambiguous sounds separated out from the /r/ cluster but at the same time excluding many of the natural /l/ segments.

Because the ambiguous sounds show a huge bias towards the natural /r/ cluster, perceptual learning should be understood as separating the ambiguous sounds from the /r/ cluster. Comparing step 0 and step 1, it is evident that perceptual learning occurs, as more than half of the 60% of the ambiguous sounds have been physically separated from the natural /r/ cluster after step 1, compared to merely around 20-30% before any retuning. This partly explains the step-like jump in Figure 12 for the Amb_l curve. Further steps of retuning keep increasing the separation up until step 4, after which the separation does not change much, and with step 4 and step 7 showing the maximum amount of separation.

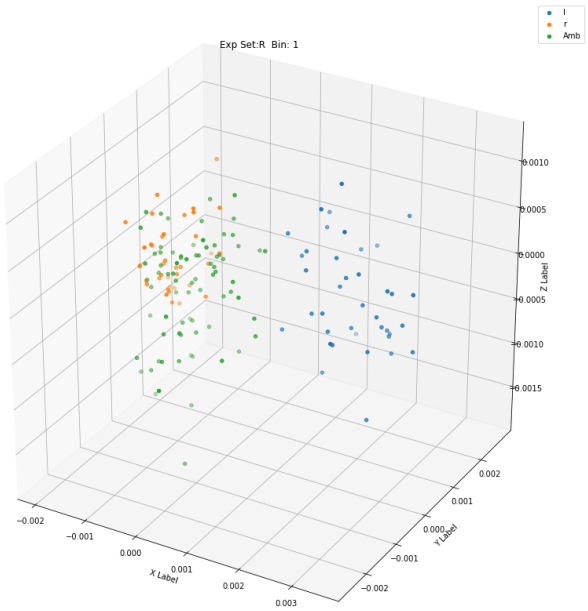
The behaviors displayed here almost matches those in Table 2, except for the fact that the clusters in Table 2 are tighter and gives a better illusion of separation between ambiguous sounds and natural /r/ sounds. This also gives us better confidence that the discussion in section 4.4 is likely to be valid.

Table 6. Visualization for the time-course of adaptation for the AmbR Bi-LSTM model (green for ambiguous [l/r], orange for /r/ and blue for /l/)

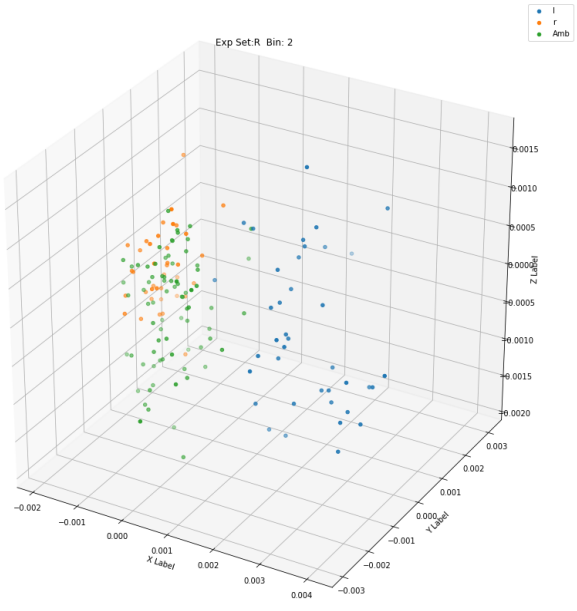
Step 0:



Step 1:



Step 2:



Step 3:

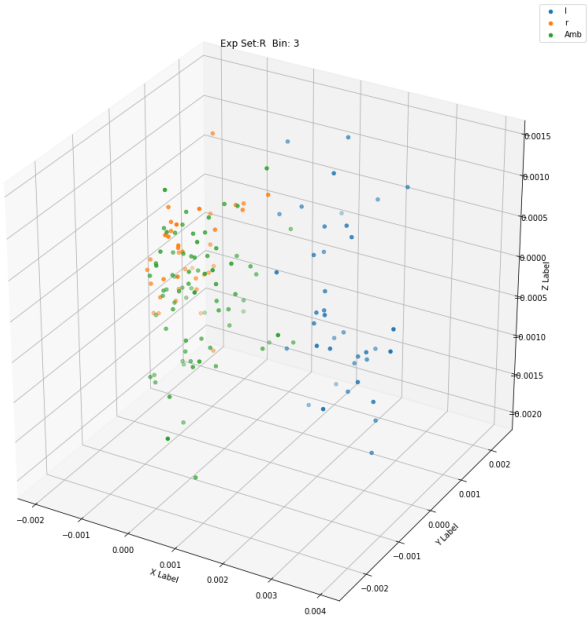


Table 6 Continued

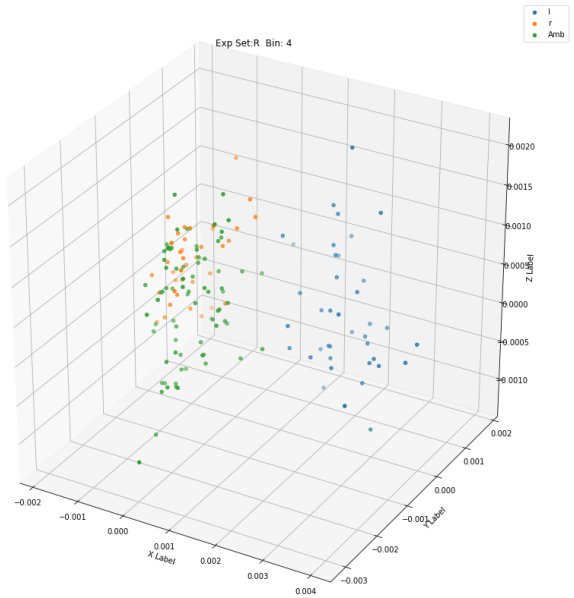
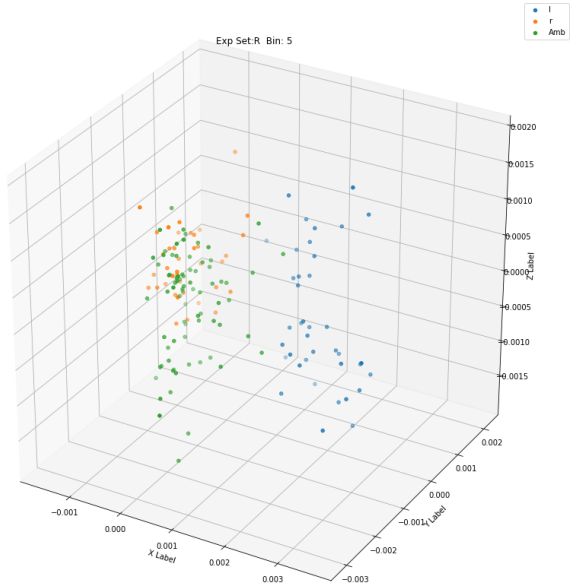
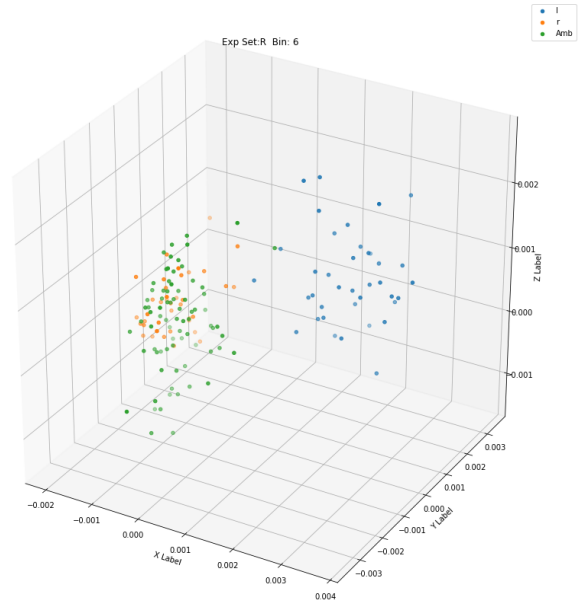
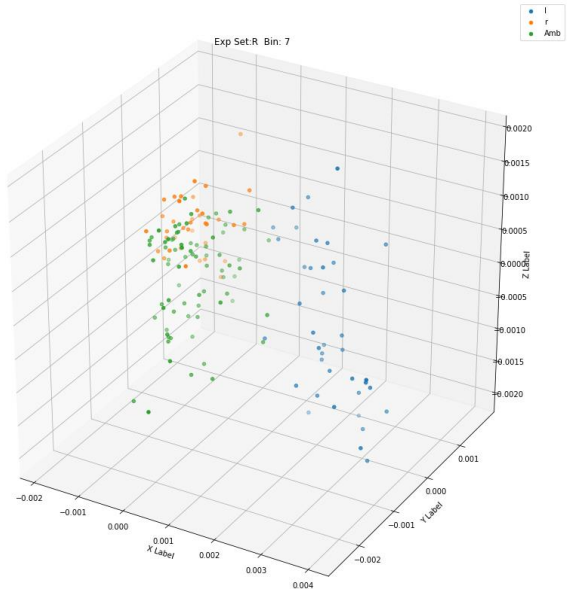
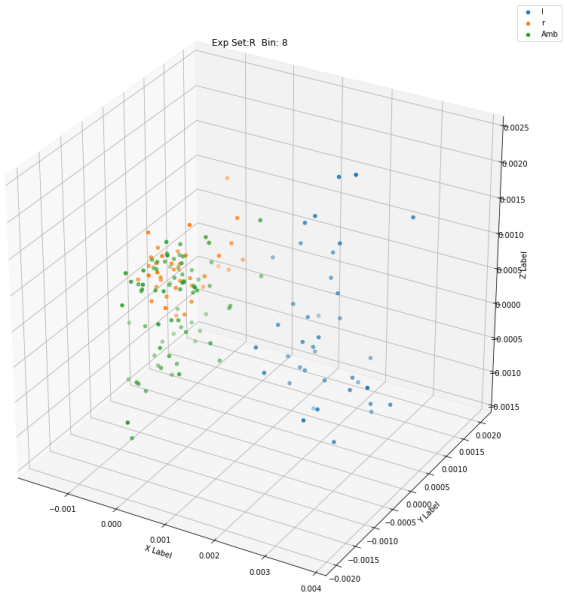
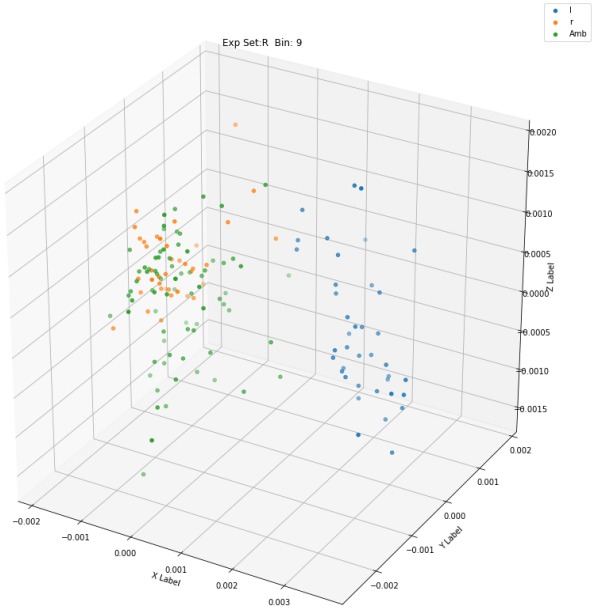
Step 4: 	Step 5: 
Step 6: 	Step 7: 

Table 6 Continued

Step 8:	Step 9:
	

For the AmbR model shown in Table 6, despite the huge initial bias of ambiguous sounds towards the natural /r/ cluster, after one step of adaptation, the perceptual learning effect is still very evident and is shown as further merging between the ambiguous sounds and natural /r/. Between step 1 and step 3, the merging continues and helps explain the gradual increase in the recognition rate seen in Figure 13 between those steps. This again agrees with the conclusions drawn in section 4.4: perceptual learning does not simply show itself as a decision boundary shift but involves a more complex process as re-learning the hidden representation of the new sounds.

The physical separation between the joint [l/r]/[r] cluster and the natural /l/ cluster also seems to increase as the model sees more retuning tokens, especially for later steps such as step 7, as compared to the control case in Table 4, although the classification rates remain somewhat constant as shown by Figure 13. Note that this is similar to the observations from Table 3 for the DNN AmbR model, and so a similar reasoning could be applied here: while this continual separation is not likely to affect recognition rates, as the separation before then is enough for the model to not confuse the ambiguous sounds as /l/, it provides proof that in later steps of adaptation, the model is still trying to strengthen its performance.

4.7 Modeling ASR as a Second Language Learner

Using the same initialization and self-training paradigm, we first tried to investigate further as to when and where the 2% slight decrease in phone error rate on the English set occurred for the model described by [110]. Using the model as described in 3.3, we first obtained the set of self-labeled transcription for the 3600 chosen utterances. 70% of the utterance with a lower phone error rate was then chosen and split evenly into ten bins. In the adaptation step, each bin was incrementally fed to the model trained on the previous bin. Unlike [110], however, the phone error rates during 10 steps of adaptation were calculated on a separate set of another 3600 utterances as opposed to the original set of utterances used for adaptation. The result is shown in Figure 14.

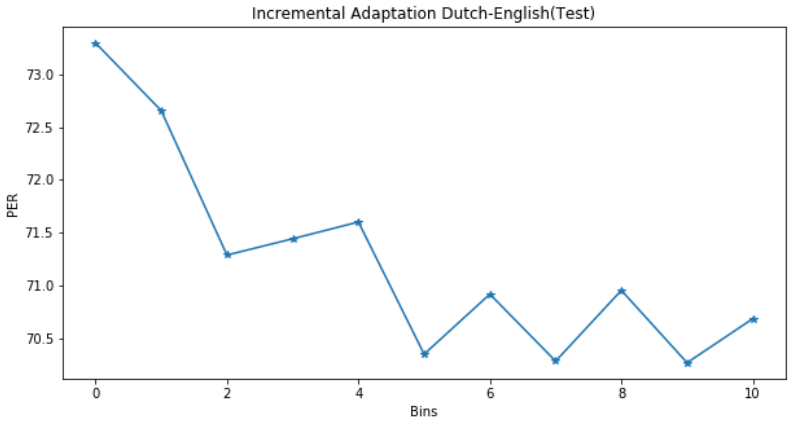


Fig. 14. Incremental adaptation of English model after initialization

According to Figure 14, the slight decrease in error rate before and after self-training mostly happens between the first two steps, corresponding to around 500 utterances. There is little to no further decrease in error rate after the fifth step, which corresponds to around 1200 utterances. Overall, the decrease in error rate was more gradual than step-like, unlike what was previously seen for the perceptual learning experiments.

In the next step, we tried to figure out where the slight decrease in error rate occurred, by plotting out the recognition rates for all the phones in the English set. Recognition rates were calculated using majority voting of CTC output within each segment as defined by the forced alignments for the utterances: a phone was counted as correctly recognized if the major vote coincided with the ground truth aligned phone. For the missing English diphthongs that were split into two phones during transcription, they were counted as recognized only if, within its forced alignment period, the CTC model output both the first phone and the second phone consecutively. Below are the results, with Figure 15 showing the recognition rates prior to self-adaptation and Figure 16 showing the recognition rates afterward.

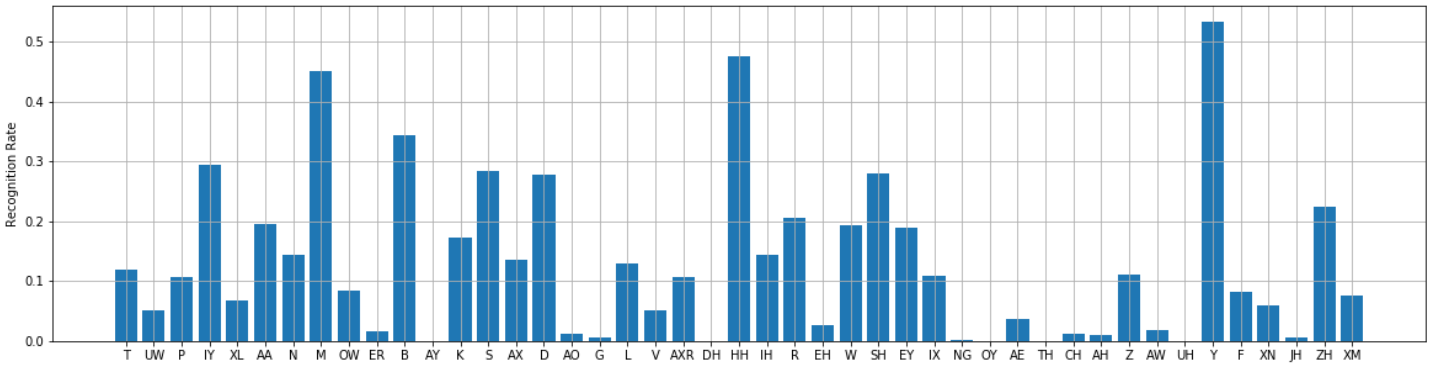


Fig. 15. Recognition rate on English phone set before self-adaptation (Step 0)

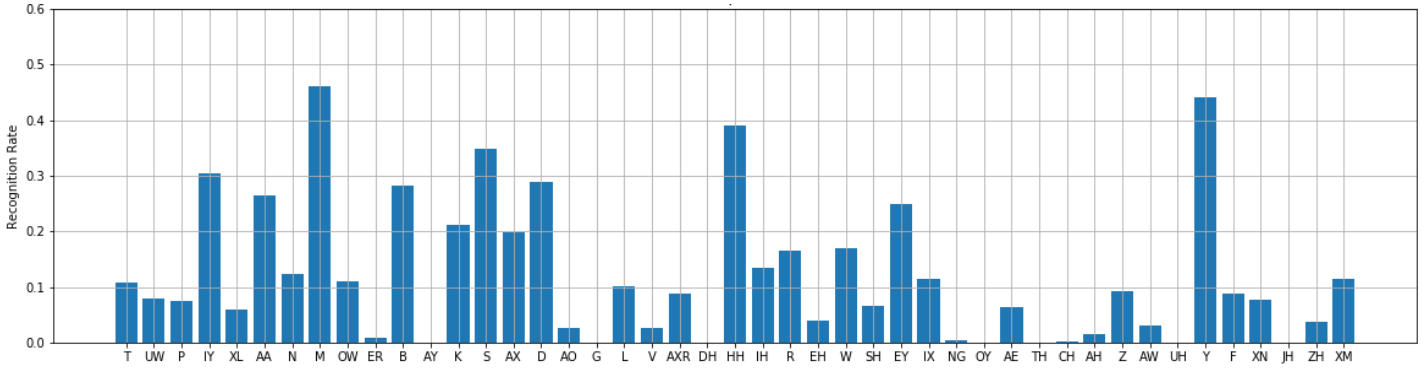


Fig. 16. Recognition rate on English phone set after self-adaptation (Step 10)

Comparing Figure 15 and Figure 16, several conclusions could be drawn:

1. It is unclear as to where the 2-3% decrease in error rate occurs. Some phones that are very well recognized before adaptation (‘HH’, Y’, ‘B’) shows a noticeable drop after the adaptation, while others (‘S’, ‘K’, ‘AX’) shows noticeable improvement. Overall the recognition rates after the adaptation are more spread-out, compared to the peak-like behavior before adaptation
2. The eight missing English diphthongs show different behaviors before and after adaptation. For example, 'JH', 'CH', 'OY','AY' are 4 out of the 8 English phones that have been separated into two phones during transcription. Their recognition rates were either zero or negligible, either before or after adaptation. The other 4 phones, 'AXR', 'XL', 'XM', 'XN' also showed a lower than average recognition rate, but their recognition rate seemed to have improved after self-adaptation.
3. The six missing English phones (‘ER’, ‘DH’, ‘AE’, ‘AH’, ‘TH’, ‘UH’) which were initialized via extrapolation between L1:1, L1:2 and L1:3 phones also showed very low recognition rates and displayed little to no change before or after initialization.

Several other bar plots were plotted to figure out the actual confusion with regard to the six missing L2 phones. Here, only the conclusions drawn from the plots are included:

'ER': On average, about 50% of the 'ER' are misclassified into 'R'. The model also tends to misclassify 10% to 15% of 'ER' into 'AX'.
'DH': 30% to 40% are classified into 'D', and 20% to 30% are classified into 'AX'.
'AE': The bar plot for 'AE' is relatively sparse. About 20% are misclassified into 'N', with another 15% into 'R' and 15% into 'AH'.
'AH': About 30% are classified into 'N', followed by 20% into 'AA', 10 to 15% into 'R', and 10% to 15% into 'NG'.
'TH': About 20% to 30% are classified into 'T', and 10% to 15% into 'D'.
'UH': The bar plot is again very sparse. 10 to 20% of 'UH' are classified into 'N', with another 10% to 20% into 'R' and another 10% to 20% into 'K'.

While some confusions displayed are reasonable, such as the high confusion rate of 'DH' into 'D', 'AH' into 'AA', and 'TH' into 'T', most of the other confusions are not, and are very unlikely to happen for a human.

The PER above was obtained by only tuning the weights with respect to softmax layers, but the weights of LSTM and convolutional layers were kept frozen. We also tried to unfreeze all the weights of the model, and got the following results shown in Table 7.

Table 7. Phone Error Rates for retuning the softmax layer only and retuning all layers

	PER After 10 Steps
Softmax Only	70.5
All Layers	69.4

Table 7 shows that, as the results obtained in [111], if all weights are unfrozen during self-adaptation, the PER further decreases by some amount (1.1%), although the amount of decrease is still unlikely to be of great significance to the acoustic model. Furthermore, unfreezing all the weights showed some amount of instability during training, which is not surprising as the model's flexibility might have been overwhelming at this early stage.

The results above have shown that even after self-adaptation, the acoustic model just was not good enough for reliably transcribing the utterances in the second language. Therefore, as discussed in section 3.3, we trained a new Dutch model with extra softmax layers for detecting articulatory features (as an auxiliary task). During self-adaptation, the model was incrementally self-trained on the articulatory feature class with the lowest error rate that had yet to be adapted on, with the transcriptions for the next step regenerated to reflect the updated model weights. The table below summarizes the error rates on the Dutch model on the CGN test set, as well as the error rates on the English model, during each step of self-labeling. Note that retuning of the current row had utilized the re-tuned weights obtained from the previous row (except for the last row, which is for comparison). The results are shown in Table 8.

Table 8. Error Rates for Phone and Articulatory Feature Class during Step-by-Step Adaptation

	Token Error Rate - Phone	Token Error Rate - Place	Token Error Rate - Manner	Token Error Rate - Voicing	Token Error Rate - Rounding	Token Error Rate - Height	Token Error Rate - Frontness
After Initialization	72.670	41.864	32.432	25.585	11.475	38.952	35.008
Rounding (Step 1)	**71.744**	**39.794**	33.626	**25.046**	**10.874**	40.089	36.646
Voicing (Step 2)	**70.828**	**39.002**	35.129	**24.850**	11.074	41.197	38.053
Manner (Step 3)	**70.197**	**38.287**	**34.090**	**24.408**	**10.727**	**40.163**	**37.202**
Frontness (Step 4)	**69.684**	**38.203**	**32.811**	24.585	11.012	40.648	37.737
Place (Step 5)	**69.260**	**38.033**	**31.942**	24.581	**10.950**	**40.494**	**37.338**
Height (Step 6)	**68.915**	**37.934**	**31.777**	25.327	11.049	40.571	37.473
Phone (Step 7)	**68.273**	38.045	32.051	25.322	**10.886**	**40.225**	**37.082**
Without using articulatory features (for comparison only)	69.396	/	/	/	/	/	/

From Table 8, some interesting phenomenon could be observed:

1. At all steps, retuning on one specific articulatory feature class results in some amount of improvement among some of the feature classes, although the feature classes that show an improvement may or may not include the feature class being adapted in that specific step (for example, the rows for frontness and height). This observation shows some inter-relatedness between different classes of articulatory features, with respect to the acoustic model learned.

2. While the specific articulatory feature classes that show an improvement change between every step in the above table, phone error rate always shows an improvement regardlessly, although the effect starts to diminish once retuning on the manner class is completed (except for the last step, as phone output layer is specifically adapted). The phone error rate improvement over the results in Table 6, as well as the fact that PER decreases even when not adapting the phone output layers specifically, provide evidence that articulatory feature detection and phone recognition tasks are very much inter-related, and that it is viable to use self-adaptation to articulatory features to guide the L2 phone learning model .

However, while 1% absolute improvement could be argued to be statistically significant ([110] pointed out that if the token error within a speech file were independent and a Bernoulli model was assumed, 0.83% of difference was enough to call two ASR models “significantly different”), the amount of learning would be still too insignificant for analyzing learning behavior. Furthermore, during the incremental adaptation on the articulatory features, it was found out that the results suffered some amount of instability, and one could easily argue that the 1% difference we obtained was an accumulation of “positive noise”.

As mentioned in section 3.3, at the time of writing, further experiments on improving the model, such as trying to learn a better phonetic boundary using an isolated word set, failed, either because of possible differences in recording environment, or unnatural segmentation effect from continuous speech. The results are therefore not discussed further here.

5. Discussion and Future Work

Inspired by the fast adaptation of human listeners to ambiguous sounds (e.g., [27, 95, 102, 109, 112, 114]), we investigated the time-course of phoneme category adaptation in a DNN, with the ultimate aim to investigate the DNN's ability to serve as a model of human perceptual learning. We based our investigation on the time-course of adaptation of the human perceptual learning experiment in [27]. In the first experiment, we provided the DNN with an increasing number of the original ambiguous acoustic stimuli from [27] as retraining tokens (in 9 bins of 4 ambiguous items), compared classification accuracy on the ambiguous items in an independent, held-out test set for the different bins, and calculated the ratio of the distance between the [l/ɹ] category and the natural [l] and [ɹ] categories, respectively, for the five hidden layers of the DNNs and for the 9 different bins. The amount of training was investigated by calculating the classification rates over 30 epochs when only one bin is used for retuning. To gain more information about the hidden representation during the phoneme category adaptation course, we visualized the phoneme category embeddings for [l], [ɹ] and [l/ɹ] after every step of incremental adaptation, using NFA and PCA.

Feeling that using a DNN that took only a fixed number of context frames for perceptual learning might not be the best model for simulating lexical guiding for humans in similar settings, we developed an end-to-end Bi-LSTM model using CTC criterion to transcribe entire utterances end-to-end. Again, we used the incremental adaptation scheme mentioned above, calculated the recognition rates for [l], [ɹ] and [l/ɹ] in a similar fashion, and performed visualizations on their phoneme category embeddings after each time step.

Results showed that, similar to human listeners, both neural network models (DNN and Bi-LSTM) quickly learned to interpret the ambiguous sound as a "natural" version of the sound. After only 4 examples of the ambiguous sound, both AmbL and AmbR models from the two vastly different architectures showed perceptual learning, and perceptual learning effect diminished for subsequent training examples, at least in terms of classification/recognition accuracy. Visualizations on the phoneme category boundaries verified this fact, as the cluster for the ambiguous sounds demonstrated a nice shift to merge with the natural /r/ cluster in the AmbR models, or at least made an effort to correct for the bias towards natural /l/ cluster for the AmbL models, from step 0 to step 1. This is in line with human lexically-guided perceptual learning; human listeners have been found to need 10-15 examples of the ambiguous sound to show the same type of step-like function [27, 102]. We should note, however, that it is not evident how to compare the 4 examples needed by the DNN with the 10-15 examples of the human listener. We know of no way to define the "learning rate" of a human listener other than by adjusting the parameters of a DNN until it matches the behavior of the human, which is an interesting avenue for further research into the DNN's ability to serve as a model of human perceptual learning. Nevertheless, both DNNs and human listeners need very little exposure to the ambiguous sound to learn to normalize it. Also, in our experiments, most significant changes

in the classification/recognition rates for both DNN and Bi-LSTM models (for the two ambiguous cases), as well as phoneme category boundaries, happened before the completion of time step 4, i.e., using less than 16 tokens (with the exception of classification rates for the DNN-AmbL model).

From the calculations of inter-category distance ratios for DNN models, we concluded that retuning took place at all levels of the DNN. In other words, retuning is not simply a change in decisions at the output layer but rather seems to be a redrawing of the phoneme category boundaries to include the ambiguous sound, via adapting the learnable weights throughout all layers of the network. From the cluster-shifting behavior during visualization for both DNN and Bi-LSTM models, we found that even at the output layer, re-tuning happens by changing/shifting the hidden representation of the ambiguous sounds so that they are more /r/ like in case of AmbR models and less /r/ like in case of AmbL models. These rather complex behaviors again were in line with what has been found for human listeners [23].

The experiments in this thesis were the first to show that, similar to inter-talker adaptation, adaptation to distorted sounds can be accomplished by a constant shift in cepstral space. Moreover, our study suggests that DNNs are more like humans than previously believed: in all cases, the DNN adapted to the deviant sound very fast and after only 4 presentations, with little or no adaptation thereafter.

However, some other future work could still be done for a more complete/rigid analysis with respect to machine perceptual learning. First, looking at the time step prior to perceptual learning, we could see that the model performed mediocrely on the natural sounds in the retuning set. This means that the model prior to perceptual learning adaptation suffered from speaker/environmental effects. Therefore, in order to fully compare with machine perceptual learning with human perceptual learning, steps must be taken to separate out the effects of adapting to new speakers/environments with the effects of adapting to ambiguous phonemes. Another possible direction includes expanding the retuning set to include training and test tokens from multiple speakers (currently, all the words in our retuning set were spoken from one single speaker). It is often found out that humans could develop perceptual learning behavior by only hearing the utterances from one single speaker, but in test stage, the tested human listener could only generalize to the utterance by that specific speaker and not to others; if multiple speakers were present, the perceptual learning effect generalizes to new speakers [16, 66, 67, 74, 83]. It would be interesting to see if machines, or neural network-based ASRs specifically, also show the same behavior with respect to speaker-generality, and if so, what are the internal mechanisms during perceptual learning of utterances from multiple speakers that help the model generalize to new speakers?

The results from the last part of this thesis, i.e., with regard to modeling neural network-based ASRs as second language learners, were not as satisfying as the results obtained previously. While we had identified the problem of baseline self-training paradigm to be that the newly added L2 phones were badly learned (or not learned at all), due to the crude acoustic model after transferring to a new language, it seemed relatively hard to find a useful criterion that could help us improve the recognition rates significantly, using the same self-training method. Utilizing

articulatory features helped a bit during self-adaptation, but the improvement was too negligible for any further formal analysis to be carried out, and proposed approach to use isolated words to help the L2 model learn better phonetic boundaries within an utterance failed due to either significantly different recording environments or unnatural word segmentation from continuous speech. Currently, at the time of writing, it remains an open question as to what modifications could be done to help the self-training L2 learning paradigm to reduce the still relatively large error rate.

It is likely, however, that the self-training paradigm, is not capable of gaining too much from self-labeled adaptation pairs if the error rate of the self-generated transcriptions is too high, which was exactly the case here. Also, it is somehow questionable if the self-training paradigm should be applied to model second language learning, as obviously, it would be just as hard to ask a human L2 learner to “self-train” on some foreign language materials, without going at least through some supervised-level learning at first.

Also, it was found out later that had normal supervised training (i.e., feed the model with speech input and ground truth transcription pairs instead of self-labeled pairs) been applied on the 3600 utterances, instead of using self-training, the model scored a 40% phone error rate on the training set, and the validation error rate (on a validation set that had the same size as the training set) was only around 5% higher than the training error rate, which showed very promising generalization effects.

Therefore, future work in the direction of studying machine L2 learning would require us to put the self-training paradigm aside for a while, and change the focus onto the following problems: at least how many hours of training utterances is needed for the L2 model to generalize well to another independent set, and what happens with the internal representation during such transfer (i.e. is it a complete remapping of phonetic space much like training an ASR from scratch, or is it some relatively simple shifts and boundary changes?) The simplest way to carry out the above would be repeating what was done for ambiguous phoneme perceptual learning experiments, i.e., splitting the adaptation set into multiple bins, and performing validation on an independent set during incremental training to see how much training data is needed for good generalization. Visualizations on phoneme category boundaries could also follow the same scheme as in section 4.6.

Another problem of interest is to figure out when in the supervised learning stage would self-training become largely beneficial. It is possible that, self-training was not so successful in our experiments due to the large phone error rate. However, it is likely that phone error rate could be greatly reduced after some amount of supervised training. Figuring out how much supervised training is needed for subsequent self-training to be successful would be another future direction to take.

6. Conclusion

This thesis focused on two aspects of modeling a neural network-based ASR as a human learner. In the first set of experiments, either a DNN or a Bi-LSTM phone recognition model was modeled as a human listener in a perceptual learning environment. Analogous to human perceptual learning experiments, we focused on when and where perceptual learning occurred in a neural network. To do this, we split our retuning set into 10 training bins, and fed our models with an increasing number of training tokens. While the context information received during training, as well as the training criterion, differed a lot for the DNN and Bi-LSTM models, the results obtained were strikingly similar: from the classification rates on the natural and ambiguous sounds across the time course, we found out that like the results from human perceptual learning experiments, both models in our experiments demonstrated a step-like response when asked to classify ambiguous tokens in the test set, after merely seeing 4 ambiguous tokens during training. The visualizations on the hidden “perceptual space” of the two types of models further validated our previous observations, as very targeted shifts of the ambiguous sound clusters were pretty evident after the first time step. Furthermore, the visualizations across the time course showed that the retuning process was not simply a phoneme boundary change but involved tuning the network weights on all levels to shift the hidden representations within the model’s “perceptual space”.

In the second set of experiments, we asked if a Dutch Bi-LSTM could learn to transcribe a second language (in this case, English). To do this, after initializing the missing phones in the second language, we asked the model to directly generate phonetic transcriptions in the first step and self-adapt to those transcriptions in the second step. We found out that using this self-training paradigm, the acoustic model remained crude after self-adaptation, and the recognition rates for the missing phones were either close to zero or very low. The self-adaptation step also showed no overall pattern of improvement in terms of recognition accuracy for each of the individual phonemes. To our surprise, however, further approaches to improve the self-training accuracy, such as self-adapting on the articulatory features to learn a better acoustic model before moving on to phone adaptation, as well as learning better phonetic boundaries using isolated words, either did not give too much of an improvement (for the articulatory features experiment), or failed due to speaker/environmental effects (for the isolated word experiment). The inability for the self-training paradigm to further learn a better acoustic model indicated that self-training, at least in the initial stage, is not likely the best candidate for second language adaptation.

References

1. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E. et. al.: Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. Proceedings of Machine Learning Research (2016)
2. Atlas, L.E., Homma, T., & Marks, R.J.: An Artificial Neural Network for Spatio-Temporal Bipolar Patterns: Application to Phoneme Classification. NIPS. (1987)
3. Bahari, M. H., Dehak, N., Van hamme, H., Burget, L., Ali, A. M., Glass, J.: Non-Negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition. In IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 22, No. 7, pp. 1117-1129 (2014)
4. Belinkov, Y., Glass, J.: Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems. ArXiv (2017)
5. Bengio, Y., Simard, P., and Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), pp. 157–166 (1994)
6. Bertelson, P., Vroomen, J., de Gelder, B.: Visual recalibration of auditory speech identification: A McGurk after effect. Psychological Science, 14, pp. 592-597 (2003)
7. Best C. T., McRoberts G. W., Sithole N. M.: Examination of perceptual reorganization for nonnative speech contrasts: Zulu click discrimination by English-speaking adults and infants. J Exp Psychol Hum Percept Perform; 4: pp. 45–60 (1988)
8. Best, C. T., McRoberts, G. W., & Goodell, E.: Discrimination of non-native consonant contrasts varying in perceptual assimilation to the listener's native phonological system. The Journal of the Acoustical Society of America, 109(2), pp. 775–794 (2001).
9. Best, C. T.: A direct realist view of cross-language speech perception. In: Strange, W., editor. Speech Perception and Linguistic Experience: Theoretical and Methodological Issues in Cross-language Speech Research. York: Timonium, MD; pp. 167-200 (1995)
10. Best, C. T.: Emergence of language-specific constraints in perception of non-native speech: A window on early phonological development. In: de Boysson-Bardies, B.; de Schonen, S.; Jusczyk, P.; MacNeilage, P.; Morton, J., editors. Developmental Neurocognition: Speech and Face Processing in the First Year. Kluwer Academic; Dordrecht, The Netherlands (1993)
11. Best, C. T.: Learning to perceive the sound pattern of English. In: Rovee-Collier, C.; Lip-sitt, LP., editors. Advances in Infancy Research. Ablex; Norwood, NJ (1994)
12. Best, C. T.: The emergence of native-language phonological influences in infants: a perceptual assimilation model. In: Nusbaum, HC., editor. The Development of Speech Perception: The Transition from Speech Sounds to Spoken Words. MIT; Cambridge, MA (1994)
13. Bishop, C. M.: Chapter 12. Continuous Latent Variables; Pattern Recognition and Machine Learning. Springer (2006)
14. Bishop, C. M.: Chapter 9. Mixture Models and EM; Pattern Recognition and Machine Learning. Springer (2006)
15. Bongaerts, T., Summeren, C., Planken, B., Schils, E.: Age and Ultimate Attainment in the Pronunciation of a Foreign Language. Studies in Second Language Acquisition. 19. pp. 447 - 465 (1997)

16. Bradlow A.R., Bent, T.: Perceptual Adaptation to Non-Native Speech. Cognition. 106(2): pp. 707–729 (2008)
17. Bradlow, A. R., Akahane-Yamada, R., Pisoni, D. B., Tohkura, Y.: Training Japanese listeners to identify English /r/ and /l/: long-term retention of learning in perception and production. Perception & psychophysics, 61(5), pp. 977–985 (1999)
18. Bradlow, A. R., Pisoni, D. B., Akahane-Yamada, R., Tohkura, Y: Training Japanese listeners to identify English /r/ and /l/: IV. Some effects of perceptual learning on speech production. The Journal of the Acoustical Society of America, 101(4), pp. 2299–2310 (1997)
19. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. in ICASSP (2016)
20. Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. ArXiv (2014)
21. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv (2014)
22. Clarke, C., Garrett, M.: Rapid adaptation to foreign-accented English. The Journal of the Acoustical Society of America. 116. pp. 3647-4658 (2005)
23. Clarke-Davidson, C. Luce, P.A., Sawusch, J.R.: Does perceptual learning in speech reflect changes in phonetic category representation or decision bias? Perception & Psychophysics, vol. 70, pp. 604-618 (2008).
24. Dahan, D., Drucker, S. J., Scarborough, R. A.: Talker adaptation in speech perception: Adjusting the signal or the representations? Cognition, 108, pp. 710-718 (2008)
25. Davis, M., Johnsrude, I., Hervais-Adelman, A., Taylor, K., Mcgettigan, C.: Lexical Information Drives Perceptual Learning of Distorted Speech: Evidence From the Comprehension of Noise-Vocoded Sentences. Journal of experimental psychology. General. 134. pp. 222-241 (2005)
26. Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-End Factor Analysis for Speaker Verification. IEEE Transactions on Audio, Speech, and Language Processing. Vol. 19 (4), pp. 788-798 (2011)
27. Drozdova, P., van Hout, R., Scharenborg, O.: Processing and adaptation to ambiguous sounds during the course of perceptual learning. Interspeech 2016, San Francisco, CA, 2811-2815 (2016).
28. Dupoux, E., Green, K.: Perceptual adjustment to highly compressed speech: effects of talker and rate changes. Journal of experimental psychology. Human perception and performance, 23 3, pp. 914-27 (1997)
29. ECE 417: Multimedia Signal Processing, Fall 2018. University of Illinois. https://courses.engr.illinois.edu/ece417/fa2018/
30. Eimas PD.: Auditory and phonetic coding of the cues for speech: Discrimination of the [r-l] distinction by young infants. Percept Psychophys 18: pp. 341–347 (1975)
31. Eisner, F., McQueen, J. M.: Perceptual learning in speech: Stability over time. Journal of the Acoustical Society of America, 119, pp. 1950-1953 (2006)
32. Fenn, K., Nusbaum, H., Margoliash, D.: Consolidation during sleep of perceptual learning of spoken language. Nature. 425. pp. 614-6 (2003)
33. Flege J. E.: Production and perception of a novel, second-language phonetic contrast. J Acoust Soc Am; 93: pp. 1589-1608 (1993)
34. Flege, J. E., Munro, M., MacKay, I.: Factors affecting degree of perceived foreign accent in a second language. The Journal of the Acoustical Society of America, pp. 3125-34 (1995).
35. Flege, J. E.: Origins and development of the Speech Learning Model. Keynote lecture presented at the 1st ASA Workshop on L2 Speech Learning, Simon Fraser Univ., Vancouver, BC (2005)
36. Flege, J. E.: Two procedures for training a novel second language phonetic contrast. Applied Psycholinguistics, 16, pp. 425-442 (1995)

37. Flege, J., Birdsong, D., Bialystok, E., Mack. M., Sung, H., Tsukada, K.: Degree of foreign accent in English sentences produced by Korean children and adults. Journal of Phonetics, 34, pp. 153-175 (2006)
38. Flege, J., Eefting, W.: The production and perception of English stops by Spanish speakers of English. Journal of Phonetics, 15, pp. 47-65 (1987)
39. Flege, J., Eefting, W.:Imitation of a VOT continuum by native speakers of English and Spanish: Evidence for phonetic category formation. Journal of the Acoustical Society of America, 83, pp. 729-740 (1988)
40. Flege, J., Port, R.: Cross-language phonetic interference: Arabic to English. Language And Speech, Vol. 24, Part 2 (1981)
41. Flege, J., Schirru, C., & MacKay, I.: Interaction between the native and second language phonetic subsystems. Speech Communication, 40, pp. 467-491 (2003)
42. Flege, J., Schmidt, A., Wharton, G.: Age of learning affects rate-dependent processing of stops in a second language. Phonetica, 53, pp. 143-161 (1996)
43. Flege, J.: Second Language Speech Learning Theory, Findings, and Problems. Speech Perception and Linguistic Experience: Issues in Cross-Language Research. (1995)
44. Flege, J.: The detection of French accent by American listeners. Journal of the Acoustical Society of America, 76, pp. 692-707 (1984)
45. Flege, J.: The production of "new" and "similar" phones in a foreign language: Evidence for the effect of equivalence classification. Journal of Phonetics, 15, pp. 47-65 (1987)
46. Flege, J.E., Yeni-Komshian, G.H., Liu, S.: Age Constraints on Second-Language Acquisition. Journal of Memory and Language, pp. 78–104 (1999)
47. Gales, M. J.: Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech & Language 12(2), pp. 75-98 (1998).
48. Gales, M., Young S.: The application of hidden markov models in speech recognition. Foundations and Trends in Signal Processing 1(3): pp. 195–304 (2007)
49. Gers, F., Schraudolph, N., and Schmidhuber, J.: Learning Precise Timing with LSTM Recurrent Networks. Journal of Machine Learning Research, 3: pp. 115–143 (2002)
50. GitHub: laurensw75/kaldi_egs_CGN. https://github.com/laurensw75/kaldi_egs_CGN
51. Goto, H.: Auditory perception by normal Japanese adults of the sounds “L” and “R”. Neuropsychologia, 9, pp. 317-323 (1971)
52. Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, US (2006).
53. Graves, A., Mohamed A., Hinton, G.: Speech Recognition with Deep Recurrent Neural Networks. International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada (2013)
54. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence. (2008)
55. Hannun, A. Y., Maas, A. L., Jurafsky, D., Ng, A. Y.: First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. ArXiv (2014)
56. Harwath, D., Glass, J.: Deep Multimodal Semantic Embeddings for Speech and Images. IEEE Automatic Speech Recognition and Understanding Workshop, pp. 237-244, Scottsdale, Arizona, US (2015)
57. Hasegawa-Johnson, M., Jyothi P., McCloy, D., Mirbagheri, M., Liberto, G., Das, A., Ekin, B., Liu, C., Manohar, V., Tang, H., Lalor, E.C., Chen, N., Hager, P., Kekona, T., Sloan, R., and Lee, A. KC: ASR for Under-Resourced Languages from Probabilistic Transcription, IEEE/ACM Trans. Audio, Speech and Language 25(1): pp. 46-59 (2017)

58. Hervais-Adelman, A., Davis M. H., Johnsrude, I. S., Carlyon R. P.: Perceptual learning of noise vocoded words: effects of feedback and lexicality. Journal of Experimental Psychology: Human Perception and Performance. 34 (2): pp. 460-74 (2008)
59. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9(8): pp.1735-1780 (1997)
60. Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, 24, pp. 417-441, 498-520. (1933)
61. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Vol. 37, pp. 448-456 (2015)
62. Iverson P., Kuhl P. K.: Influences of phonetic identification and category goodness on American listeners' perception of /r/ and /l/. J Acoust Soc Am; 99(2): pp.1130-40 (1996)
63. Jolliffe, I.T.: Principal Component Analysis. Springer (2002)
64. Karaminis, T., Scharenborg, O.: The effects of background noise on native and non-native spoken-word recognition: A computational modelling approach. Proceedings of the Cognitive Science conference, Madison, WI, USA (2018).
65. Kauderer-Abrams, E.: Quantifying Translation-Invariance in Convolutional Neural Networks. ArXiv (2018)
66. Kraljic, T., & Samuel, A. G.: Generalization in perceptual learning for speech. Psychonomic Bulletin & Review, 13, pp. 262-268 (2006)
67. Kraljic, T., & Samuel, A. G.: Perceptual adjustments to multiple speakers. Journal of Memory & Language, 56, pp. 1-15 (2007)
68. Kraljic, T., Brennan, S. E., & Samuel, A. G.: Accommodating variation: Dialects, idiolects, and speech processing. Cognition,107, pp. 54-81 (2008)
69. Kraljic, T., Samuel, A. G., Brennan, S. E.: First impressions and last resorts: How listeners adjust to speaker variability. Psy-chological Science, 19, pp. 332-338 (2008)
70. Kraljic, T., Samuel, A. G.: Perceptual learning for speech: Is there a return to normal? Cognitive Psychology, 51, pp. 141-178 (2005)
71. Kuhl P. K., Williams K. A., Lacerda F., Stevens K. N., Lindblom B.: Linguistic experience alters phonetic perception in infants by 6 months of age. Science, 255: pp. 606-608 (1992)
72. Kuhl P. K.: Human adults and human infants show a "perceptual magnet effect" for the prototypes of speech categories, monkeys do not. Percept Psychophys; 50: pp. 93-107 (1991)
73. Kuhl P. K.: Innate predispositions and the effects of experience in speech perception: the native language magnet theory. In Developmental neurocognition: speech and face processing in the first year of life. Edited by de Boysson-Bardies B, de Schonen S, Jusczyk P, McNeilage P, Morton I. Dordrecht, Netherlands: Kluwer Academic Publishers; pp. 259-274 (1993)
74. Kuhl, P.: Learning and representation in speech and language. Current Opinion in Neurobiology, vol. 4, pp. 812–822 (1994).
75. Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., Bengio, Y.: Batch normalized recurrent neural networks. ArXiv (2015)
76. Leach, L., & Samuel, A. G.: Lexical configuration and lexical engagement: When adults learn new words. Cognitive Psychology, 55, pp. 306-353 (2007)
77. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Computer Vision and Pattern Recognition pp. 105-114 (2017)
78. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, pp. 788–791 (1999)
79. Lenneberg, E. H.: On explaining language. Science 164.3880 pp. 635-643 (1969)

80. Liao, H.: Speaker adaptation of context dependent deep neural networks. Proceedings of ICASSP, pp. 7947-7951 (2013).
81. Liberman, M., et al. TI 46-Word LDC93S9. Web Download. Philadelphia: Linguistic Data Consortium (1993)
82. Lively S. E., Pisoni D. B., Yamada R. A., Tohkura Y., Yamada T.: Training Japanese listeners to identify English /r/ and /l/. III. Long-term retention of new phonetic categories. The Journal of the Acoustical Society of America, pp. 2076–2087 (1994)
83. Lively, S. E., Logan, J. S., Pisoni, D. B.:Training Japanese listeners to identify English /r/ and /l/. II: The role of phonetic environment and talker variability in learning new perceptual categories.The Journal of the Acoustical Society of America, pp. 1242–1255 (1993)
84. Logan, J. S., Lively, S. E., Pisoni, D. B.: Training Japanese listeners to identify English /r/ and /l/: A first report. The Journal of the Acoustical Society of America, pp. 874–886 (1991)
85. Luong, T., Pham, H., Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing pp. 1412–1421 (2015)
86. Maye, J., Aslin, R. N., Tanenhaus, M. K.: The weckud wetch of the wast: Lexical adaptation to a novel accent. Cognitive Science,32, pp. 543-562 (2008)
87. Maye, J., Weiss, D., Aslin, R.: Statistical phonetic learning in infants: Facilitation and feature generalization. Developmental science. 11. pp. 122-34 (2008)
88. McAllister, R., Flege, J., Piske, T.: The influence of the L1 on the acquisition of Swedish vowel quantity by native speakers of Spanish, English and Estonian. Journal of Phonetics, 30, pp. 229-258 (2002)
89. McGarr, N.S.: The Intelligibility of Deaf Speech to Experienced and Inexperienced Listeners. Journal of Speech, Language, and Hearing Research. Vol. 26 (3), pp. 451-458 (1983)
90. McQueen, J. M., Cutler, A., Norris, D.: Phonological ab-straction in the mental lexicon. Cognitive Science, 30, pp. 1113-1126 (2006)
91. McQueen, J. M., Norris, D., & Cutler, A.: The dynamic nature of speech perception. Language & Speech, 49, pp. 101-112 (2006)
92. Miao, Y., Gowayyed, M., Metze, F.: EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding. In Proc. Automatic Speech Recognition and Understanding Workshop (ASRU), Scottsdale, AZ; U.S.A. (2015)
93. Mohri, M., Pereira, F., Riley M.: Weighted finite-state transducers in speech recognition. Computer Speech and Language. Vol. 20, No. 1, pp. 69–88 (2002)
94. Munro, M., Flege, J., & MacKay, I.: The effect of age of second-language learning on the production of English vowels. Applied Psycholinguistics, 17, pp. 313-334 (1996)
95. Norris, D., McQueen, J. M., Cutler, A.: Perceptual learning in speech. Cognitive Psychology 47, 204-238 (2003).
96. Nygaard, L. C., Pisoni, D. B.: Talker-specific learning in speech perception. Perception & Psychophysics, 60 (3), pp. 355-376 (1998)
97. Oostdijk, N.H.J., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M., Baayen, H.: Experiences from the Spoken Dutch Corpus project. Proc. LREC – Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, pp. 340-347 (2002).
98. Pallier, C., Sebastian Galles, N., Dupoux, E., Christophe, A., Mehler, J.: Perceptual adjustment to time-compressed speech: A cross-linguistic study. Memory & Cognition. 26. pp. 844-851 (1998)
99. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on International Conference on Machine Learning - Vol. 28, pp. 1310-1318 (2013)

100. Pearson, K.: On lines and planes of closest fit to systems of points in space. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series 2, pp. 559–572 (1901)
101. Pitz, M., Ney, H.: Vocal Tract Normalization Equals Linear Transformation in Cepstral Space. IEEE Transactions on Speech and Audio Processing 13(5):930-944 (2005).
102. Poellmann, K., McQueen, J.M., Mitterer, H.: The time course of perceptual learning. Proceedings of ICPhS (2011).
103. Polka L., Werker J. F.: Developmental changes in perception of nonnative vowel contrasts. J Exp Psych: Hum Percept Perform; 20: pp. 421-435 (1994)
104. Polka, L.: Cross-language speech perception in adults: Phonemic, phonetic, and acoustic contributions. Jornal of the Acoustical Society of America, 89, pp. 2961-2977 (1991)
105. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi Speech Recognition Toolkit. IEEE Workshop on Automatic Speech Recognition and Understanding, Hawaii, US (2011).
106. Redmon, J., Farhadi A.: Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (2017)
107. Repp, B. H.: Categorical Perception: Issues, Methods, Findings. Speech and Language, Volume 10, pp. 243-335 (1984)
108. Reynolds, D., Quatieri, T. F., Dunn, R. B.: Speaker Verification using Adapted Gaussian Mixture Models. Digital Signal Processing, Vol. 10, pp. 19–41 (2000)
109. Samuel, A. G., Kraljic, T.: Perceptual learning in speech perception. Attention, Perception & Psychophysics 71, 1207-1218 (2009).
110. Scharenborg, O., Ciannella, F., Palaskar, S., Black, A., Metze, F., Ondel, L., Hasegawa-Johnson, M.: Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system: Preliminary results. Proceedings of the International Conference on Natural Language, Signal and Speech Processing, Casablanca, Morocco (2017)
111. Scharenborg, O., Ebescharl, P., Ciannella, F., Hasegawa-Johnson, M., Dehak, N.: Building an ASR system for Mboshi using a cross-language definition of acoustic units approach. Proceedings of the International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU'18), 167-171, Gurugram, India (2018)
112. Scharenborg, O., Janse, E: Comparing lexically-guided perceptual learning in younger and older listeners. Attention, Perception, and Psychophysics 75 (3), 525-536 (2013). doi: 10.3758/s13414-013-0422-4.
113. Scharenborg, O., Tiesmeyer, S., Hasegawa-Johnson, M., Dehak, N.: Visualizing phoneme category adaptation in deep neural networks. Proceedings of Interspeech, Hyderabad, India (2018).
114. Scharenborg, O., Weber, A., Janse, E.: The role of attentional abilities in lexically-guided perceptual learning by older listeners. Attention, Perception, & Psychophysics 77 (2), 493–507 (2015). https://doi.org/10.3758/s13414-014-0792-2
115. Scharenborg, O.: Modeling the use of durational information in human spoken-word recognition. Journal of the Acoustical Society of America, 127 (6), 3758-3770 (2010).
116. Schuster, M., Paliwal, K. K.: Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing, Vol. 45, pp. 2673–2681 (1997)
117. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (2015)
118. Stacey, P.C., Summerfield, A.Q.: Effectiveness of computer-based auditory training in improving the perception of noise-vocoded speech. The Journal of the Acoustical Society of America, 121 5 Pt1, pp. 2923-2935 (2007)

119. Trehub, S. E.: The discrimination of foreign speech contrasts by adults and infants. Child Development, 47, pp. 466-472 (1976)
120. Vroomen, J., van Linden, S., de Gelder, B., Bertelson, P.: Visual recalibration and selective adaptation in auditory-visual speech perception: Contrasting build-up courses. Neuropsychologia, 45, pp. 572-577 (2007)
121. Wang, Y., Spence, M. M., Jongman, A., Sereno, J. A.: Training American listeners to perceive Mandarin tones. Journal of the Acoustical Society of America, 106, pp. 3649-3658 (1999)
122. Wardhaugh, R.: The Contrastive Analysis Hypothesis. Teachers of English to Speakers of Other Languages, Inc. (TESOL) Vol. 4, No. 2, pp. 123-130 (1970)
123. Werker IF, Polka L: The ontogeny and developmental significance of language-specific phonetic perception. In Developmental neurocognition: speech and face processing in the first year of life. Edited by de Boysson-Bardies B, de Schonen S, Jusczyk P, McNeilage P, Morton J. Dordrecht, Netherlands: Kluwer Academic Publishers; pp. 275-288 (1993)
124. Yeni-Komshian, G., Flege, J. E., Liu, S.: Pronunciation proficiency in first and second languages of Korean–English bilinguals. Bilingualism: Language and Cognition, 3, pp. 131–149 (2000)
125. Yosinski, J., Clune, J., Nguyen, A.M., Fuchs, T.J., Lipson, H.: Understanding Neural Networks Through Deep Visualization. ArXiv (2015)