



NIESR: Nuisance Invariant End-to-end Speech Recognition

I-Hung Hsu, Ayush Jaiswal, Premkumar Natarajan

USC Information Sciences Institute, Marina del Rey, CA, USA

{ihunghsu, ajaiswal, pnataraj}@isi.edu

Abstract

Deep neural network models for speech recognition have achieved great success recently, but they can learn incorrect associations between the target and nuisance factors of speech (e.g., speaker identities, background noise, etc.), which can lead to overfitting. While several methods have been proposed to tackle this problem, existing methods incorporate additional information about nuisance factors during training to develop invariant models. However, enumeration of all possible nuisance factors in speech data and the collection of their annotations is difficult and expensive. We present a robust training scheme for end-to-end speech recognition that adopts an unsupervised adversarial invariance induction framework to separate out essential factors for speech-recognition from nuisances without using any supplementary labels besides the transcriptions. Experiments show that the speech recognition model trained with the proposed training scheme achieves relative improvements of 5.48% on WSJ0, 6.16% on CHiME3, and 6.61% on TIMIT dataset over the base model. Additionally, the proposed method achieves a relative improvement of 14.44% on the combined WSJ0+CHiME3 dataset.

Index Terms: invariant representation learning, speech recognition, adversarial learning

1. Introduction

With the aid of recent advances in neural networks, end-to-end deep learning systems for automatic speech recognition (ASR) have gained popularity and achieved extraordinary performance on a variety of benchmarks [1, 2, 3, 4]. End-to-end ASR models typically consist of Recurrent Neural Networks (RNNs) with Sequence-to-Sequence (Seq2Seq) architectures and attention mechanisms [5], RNN transducers [6], or transformer networks [3]. These systems learn a direct mapping from an audio signal sequence to a sequence of text transcriptions. However, the input audio sequence often contains nuisance factors that are irrelevant to the recognition task and the trained model can incorrectly learn to associate some of these factors with target variables, which leads to overfitting. For example, besides linguistic content, speech data contains nuisance information about speaker identities, background noise, etc., which can hurt the recognition performance if the distributions of these attributes are mismatched between training and testing.

A common method for combatting the vulnerability of deep neural networks to nuisance factors is the incorporation of invariance induction during model training. For example, invariant deep models have achieved considerable success in computer vision [7, 8, 9] and speech recognition [10, 11, 12, 13]. Serdyuk et al. [10] obtain noise-invariant representations by employing noise-condition annotations and the gradient reversal layer [14] for acoustic modeling. Similarly, Meng et al. [11] utilize speaker information to train a speaker-invariant model for senone prediction. Hsu et al. [12] extract domain-invariant

features using a factorized hierarchical variational autoencoder. Liang et al. [13] force their end-to-end ASR model to learn similar representations for clean input instances and their synthetically generated noisy counterparts.

While these methods work well at handling discrepancies between training and testing datasets for ASR systems, they require domain knowledge [12], supplementary nuisance information during training (e.g., speaker identities [11], recording environments [10], etc.), or pairwise data [13]. However, these requirements are difficult and expensive to fulfill in real world, e.g., it is hard to enumerate all possible nuisance factors and collect corresponding annotations.

In this work, we propose a new training scheme, namely NIESR, which adopts the unsupervised adversarial invariance learning framework (UAI) [7] for end-to-end speech recognition. Without incorporating supervised information of nuisances for the input signal features, the proposed method is capable of separating the underlying elements of speech data into two series of latent embeddings – one containing all the information that is essential for ASR, and the other containing information that is irrelevant to the recognition task (e.g. accents, background noises, etc.). Experimental results show that the proposed training method boosts the end-to-end ASR performance on WSJ0, CHiME3, and TIMIT datasets. We also show the effectiveness of combining NIESR with data augmentation.

2. Methodology

In this section, we present the proposed NIESR model for nuisance-invariant end-to-end speech recognition, where the invariance is achieved by adopting the UAI framework [7]. We begin by describing the base Seq2Seq ASR model. Subsequently, we introduce the UAI framework for unsupervised adversarial invariance induction. Finally, we present the complete design of the proposed NIESR model.

2.1. Base Sequence-to-sequence Model

We are interested in learning a mapping from a sequence of acoustic spectra features $\mathbf{x} = (x_1, x_2, \dots, x_T)$ to a series of textual characters $\mathbf{y} = (y_1, y_2, \dots, y_S)$, given a dataset $D \equiv \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^N$, following the formulation of Chan et al. [5]. We employ a Seq2Seq model for this task, which estimates the probability of each character output y_i by conditioning over the previous characters $\mathbf{y}_{1:(i-1)}$ and the input sequence \mathbf{x} . Thus, the conditional probability of the entire output \mathbf{y} is:

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}, \mathbf{y}_{1:(i-1)}) \quad (1)$$

A Seq2Seq model is composed of two modules: an encoder *Enc* and a decoder *Dec*. *Enc* transforms the input features \mathbf{x} into a high-level representation $\mathbf{h} = (h_1, h_2, \dots, h_T)$, i.e. $\mathbf{h} = \text{Enc}(\mathbf{x})$ and *Dec* infers the output sequence \mathbf{y} from \mathbf{h} . We

model *Enc* as a stack of Bidirectional Long-Short Term Memory (BLSTM) layers with interspersed projected-subsampling layers [15]. The subsampling layer projects a pair of consecutive input frames (u_{2i-1}, u_{2i}) to a single lower-dimensional frame v_i . We model *Dec* as an attention-based LSTM transducer [16], which employs \mathbf{h} to produce the output character sequence. At every time step, *Dec* generates a probability distribution of y_i over character sequences, which is a function of a transducer state s_i and an attention context c_i . We denote this function as CharDist, which is implemented as a single layer perceptron with softmax activation:

$$s_i = \text{LSTM}([y_{i-1}, c_{i-1}], s_{i-1}) \quad (2)$$

$$p(y_i | \mathbf{x}, \mathbf{y}_{1:(i-1)}) = \text{CharDist}(s_i, c_i) \quad (3)$$

In order to calculate the attention context c_i , we employ the hybrid location-aware content-based attention mechanism proposed by [17]. Specifically, the attention energy $e_{i,j}$ for frame j at time-step i takes previous attention alignment α_{i-1} into account through the convolution operation:

$$e_{i,j} = w^\top \tanh(Ws_i + Vh_j + U(F * \alpha_{i-1}) + b) \quad (4)$$

where w, b, W, V, U , and F are learned parameters and $*$ depicts the convolution operation. The attention alignment $\alpha_{i,j}$ and the attention context c_i is then calculated as:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^L \exp(e_{i,k})}, \quad c_i = \sum_{j=1}^L \alpha_{i,j} h_j \quad (5)$$

The base model is trained by minimizing the cross-entropy loss:

$$L_y = - \sum_i \log p(y_i | \mathbf{x}, \mathbf{y}_{1:(i-1)}) \quad (6)$$

2.2. Unsupervised Adversarial Invariance Induction

Deep neural networks (DNNs) often learn incorrect associations between nuisance factors in the raw data and the final target, leading to poor generalization [7]. In the case of ASR, the network can link accents, speaker-specific information, or background noise with the transcriptions, resulting in overfitting. In order to cope with this issue, we adopt the unsupervised adversarial invariance (UAI) [7] framework for learning invariant representations that eliminate factors irrelevant to the recognition task without requiring any knowledge of nuisance factors.

The working principle of UAI is to learn a split representation of data as \mathbf{h}^1 and \mathbf{h}^2 , where \mathbf{h}^1 contains information relevant to the prediction task (here ASR) and \mathbf{h}^2 holds all other information about the input data. The underlying mechanism for learning such a split representation is to induce competition between the main prediction task and an auxiliary task of data reconstruction. In order to achieve this, the framework uses \mathbf{h}^1 for the prediction task and a noisy version $\tilde{\mathbf{h}}^1$ of \mathbf{h}^1 along with \mathbf{h}^2 for reconstruction. In addition, a disentanglement constraint enforces that \mathbf{h}^1 and \mathbf{h}^2 contain independent information. The prediction task tries to pull relevant factors into \mathbf{h}^1 , while the reconstruction task drives \mathbf{h}^2 to store all the information about input data because $\tilde{\mathbf{h}}^1$ is unreliable. However, the disentanglement constraint forces the two embeddings to not contain overlapping information, thus leading to competition. At convergence, this results in a nuisance-free \mathbf{h}^1 that contains only those factors that are essential for the prediction task.

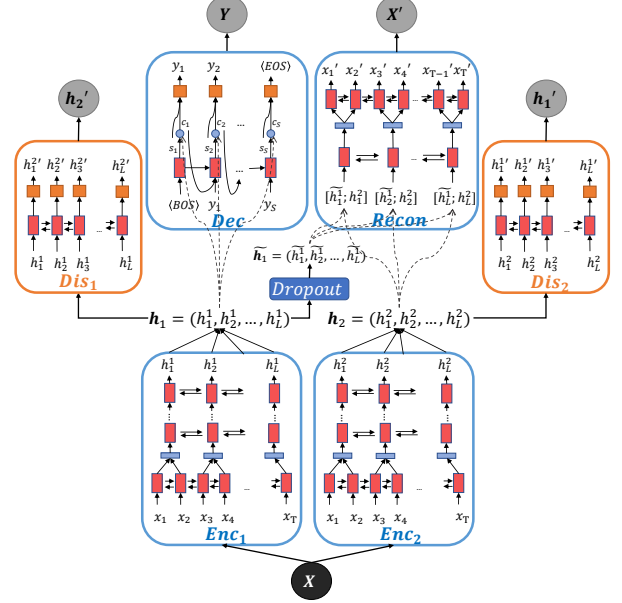


Figure 1: NIESR: The two encoders Enc_1 and Enc_2 are BLSTM-based feature extractors that encode the input sequence \mathbf{x} into representations \mathbf{h}^1 and \mathbf{h}^2 . The two encodings are disentangled by adversarially training the two disentanglers, Dis_1 and Dis_2 , which aim to predict one embedding from another. Dec is an attention-based decoder that generates the target \mathbf{y} character from \mathbf{h}^1 . $Recon$ is a BLSTM-based reconstructor that decodes \mathbf{h}^2 and the noisy $\tilde{\mathbf{h}}^1$ back to the input-sequence \mathbf{x}

2.3. NIESR Model Design and Optimization

The NIESR model comprises five types of modules: (1) encoders Enc_1 and Enc_2 that map input data to the encodings \mathbf{h}^1 and \mathbf{h}^2 , respectively, (2) a decoder Dec that infers target \mathbf{y} from \mathbf{h}^1 , (3) a dropout layer that converts \mathbf{h}^1 into its noisy version $\tilde{\mathbf{h}}^1$, (4) a reconstructor $Recon$ that reconstructs input data from $[\tilde{\mathbf{h}}^1, \mathbf{h}^2]$, and (5) two adversarial disentanglers Dis_1 and Dis_2 that try to infer each embedding (\mathbf{h}^1 or \mathbf{h}^2) from the other. Figure 1 shows the complete NIESR model.

The encoder Enc_1 and decoder Dec follow the base model design as described in Section 2.1, i.e., an attention-based Seq2Seq model for the speech recognition task. Enc_2 is designed to have exactly the same structure as Enc_1 . The dropout layer is introduced to make $\tilde{\mathbf{h}}^1$ an unreliable source of information for reconstruction, which influences the reconstruction task to extract all information about \mathbf{x} into \mathbf{h}^2 [7]. $Recon$ is modeled as a stack of BLSTM layers interspersed with novel upsampling layers, which perform decompression by splitting information in each time-frame to two frames. This is the inverse of the subsampling layers [15] used in Enc_1 and Enc_2 . The upsampling operation is formulated as:

$$[u_{2i-1}, u_{2i}] = \text{BLSTM}([\tilde{h}_i^1, h_i^2], s_{i-1}) \quad (7)$$

$$o_{2i} = P u_{2i}, \quad o_{2i-1} = P u_{2i-1} \quad (8)$$

where $[\cdot, \cdot]$ represents concatenation, o is the output, and P is a learned projection matrix.

The adversarial disentanglers Dis_1 and Dis_2 model the UAI disentanglement constraint discussed in Section 2.2 following previous works [7, 8, 9]. Dis_1 tries to predict \mathbf{h}^2 from

\mathbf{h}^1 and Dis_2 tries to do the inverse. This is directly opposite to the desired independence between \mathbf{h}^1 and \mathbf{h}^2 . Thus, training Dis_1 and Dis_2 adversarially against the rest of the model helps achieve the independence goal. Unlike previous works [7, 8, 9], the encodings \mathbf{h}^1 and \mathbf{h}^2 for this work are vector-sequences instead of single vectors: $\mathbf{h}^1 = (h_1^1, h_2^1, \dots, h_L^1)$ and $\mathbf{h}^2 = (h_1^2, h_2^2, \dots, h_L^2)$. Naïve instantiations of the disentangled would perform frame-specific predictions of h_i^2 from h_i^1 and vice versa. However, each pair of h_i^1 and h_i^2 generated at the time-step i contains information not only from frame i but also from other frames across the time-span. This is because Enc_1 and Enc_2 are modeled as RNNs. Therefore, a better method to perform disentanglement for sequential representations is to use the whole series of \mathbf{h}^1 or \mathbf{h}^2 to estimate every element of the other. Hence, we model Dis_1 and Dis_2 as BLSTMs.

The proposed NIESR model is optimized by adopting the UAI training strategy [7, 9], i.e., playing a game where we treat Enc_1 , Enc_2 , Dec , and $Recon$ as one player \mathbf{P}_1 , and Dis_1 and Dis_2 as the other player \mathbf{P}_2 . The model is trained using a scheduled update scheme where we freeze the weights of one player model when we update the weights of the other. The training objective comprises three tasks: (1) predicting transcriptions from the input signal, (2) reconstruction of the input, and (3) adversarial prediction of each of \mathbf{h}^1 and \mathbf{h}^2 from the other. The objective of the first task is written as Equation 6. The goal for the reconstruction task is to minimize the mean squared error (MSE) between \mathbf{x} and the reconstructed \mathbf{x}' :

$$L_x = \text{MSE}(Recon([\psi(Enc_1(\mathbf{x})), Enc_2(\mathbf{x})]), \mathbf{x}) \quad (9)$$

where ψ means dropout. The training objective for the disentangled is to minimize the MSE between embeddings predicted by the disentanglers and the embeddings generated from the encoder. However, that of the encoders is to generate \mathbf{h}^1 and \mathbf{h}^2 that are not predictive of each other. Hence, in the scheduled update scheme, the targets \mathbf{t}^1 and \mathbf{t}^2 for the disentanglers are different when updating the player models P_1 versus P_2 , following [9]. The loss can be written as:

$$L_d = \text{MSE}(Dis_1(Enc_1(\mathbf{x})), \mathbf{t}^1) \quad (10)$$

$$+ \text{MSE}(Dis_2(Enc_2(\mathbf{x})), \mathbf{t}^2) \quad (11)$$

where \mathbf{t}^1 and \mathbf{t}^2 are set as \mathbf{h}^2 and \mathbf{h}^1 , respectively, when updating \mathbf{P}_2 but are set to random vectors when updating \mathbf{P}_1 .

Overall, the model is trained through backpropagation by optimizing the objective described in Equation 12, where the loss-weights α , β , and γ are hyperparameters, which are decided by the performance on the development set.

$$L = \alpha L_y + \beta L_x + \gamma L_d \quad (12)$$

Inference with NIESR involves a forward pass of data through Enc_1 followed by Dec . Hence, the usage and computational cost of NIESR for inference is the same as the base model.

3. Experiments

The effectiveness of NIESR is quantified through the performance improvement achieved by adopting the invariant learning framework. We provide experimental results on speech recognition on three benchmark datasets: the Wall Street Journal Corpus (WSJ0) [18], CHiME3 [19], and TIMIT [20]. We additionally provide results on the combined WSJ0+CHiME3 dataset.

Table 1: Hyperparameters for the base model.

Item	Setting
<i>Enc</i> and <i>Dec</i> LSTM Dimension	200
Subsampling Projected Dimension	200
Attention Dimension	200
Attention Convolution Channel	10
Attention Convolution Kernel Size	100
Optimizer	Adam
Learning Rate	5e-4

3.1. Datasets

WSJ0: This dataset is a collection of readings of the Wall Street Journal. It contains 7,138 utterances in the training set, 410 in the development set, and 330 in the test set. We use 40-dimensional log Mel filterbank features as the model input, and normalize the transcriptions to capitalized character sequences. **CHiME3:** CHiME3 dataset contains: (1) WSJ0 sentences spoken in challenging noisy environments (real data) and (2) WSJ0 readings mixed with four different background noise (simulated data). The real speech data was recorded in five noisy environments using a six-channel tablet-based microphone array. Training data consists of 1,999 real noisy utterances from four speakers, and 7,138 simulated noisy utterances from 83 speakers in the WSJ0 training set. In total, there are 3,280 utterances in the development set, and 2,640 utterances in the test set containing both real and simulated data. The speakers in training, development, and test set are mutually different. In our experiments, we follow [11] to use far-field speech from the fifth microphone channel for all sets. We adopt the same input-output setting for CHiME3 as WSJ0.

TIMIT: This corpus contains a total of 6,300 sentences, with 10 sentences spoken by 630 speakers each with 8 different dialects. Among them, utterances from 168 different speakers are held-out as the test set. We further select sentences from 4 speakers of each dialect group, i.e., 32 speakers in total, from the remaining data to form the development set. Thus, all speakers in training, development, and test sets are different. Models were trained on 80 log Mel filterbank features and capitalized character sequences were treated as targets.

3.2. Experiment Setup

We train the base model without using invariance induction, i.e., the model consisting of *Enc* and *Dec* (Section 2.1), as a baseline. We feed the whole sequence of spectra features to *Enc* and get the predicted character sequence from *Dec*. We use a stack of two BLSTMs with a subsampling layer (as described in Section 2.1) in between for *Enc*. *Dec* is implemented as a single layer LSTM combined with attention modules introduced in Section 2.1. All the models were trained with early stopping with 30 epochs of patience and the best model is selected based on the performance on the development set. Other model and training hyperparameters are listed in Table 1.

We augment the base model with *Enc_2*, *Recon*, *Dis_1*, and *Dis_2*, while treating *Enc* as *Enc_1*, to form the NIESR model. *Enc_2* has the same hyperparameter setting and structure as *Enc_1*. *Recon* is modeled as a cascade of a BLSTM layer, an upsampling layer, and another BLSTM layer. *Dis_1* and *Dis_2* are implemented as BLSTMs followed by two fully-connected layers. We update the player models \mathbf{P}_1 and \mathbf{P}_2 in the frequency ratio of 1 : 5 in our experiments. Hyperparameters for *Enc_1* and *Dec* are the same as the base model. Additional hyperparameters for NIESR are summarized in Table 2.

Table 2: Hyperparameters for the NIESR model.

Item	Setting
Recon LSTM Dimension	300
Upsampling Projected Dimension	200
Dis_1, Dis_2 Dimension	200
Dropout layer rate	0.4
Optimizer	Adam
Learning Rate for P_1	5e-4
Learning Rate for P_2	1e-3
α, β, γ for WSJ0	100, 10, 1
α, β, γ for CHiME3	100, 1, 0.5
α, β, γ for TIMIT	100, 50, 1

Table 3: Speech recognition performance as CER (%). Values in parentheses show relative improvement (%) over Base model.

Model	WSJ0	CHiME3	TIMIT
Base	12.95	44.61	28.76
Spk-Inv	12.31 (4.94)	43.93 (1.52)	28.45 (1.08)
Env-Inv	–	42.61 (4.48)	–
Dial-Inv	–	–	28.29 (1.63)
NIESR	12.24 (5.48)	41.86 (6.16)	26.86 (6.61)

We further provide results of a stronger baseline model that utilizes *labeled* nuisances z (speakers for WSJ0, speakers and noise environment condition for CHiME3, speakers and dialect groups for TIMIT) with the gradient reversal layer (GRL) [14] to learn invariant representations. Specifically, the model consists of *Enc*, *Dec*, and a classifier with a GRL between the embedding learned from *Enc* and the classifier, following the standard setup in [14]. The target for the classifier is to predict z from the embedding while the direction of the training gradient to *Enc* is flipped. We denote this model as **Spk-Inv** for speaker-invariance, **Env-Inv** for environment-invariance in CHiME3, and **Dial-Inv** for dialect-invariance in TIMIT.

3.3. ASR Performance on Benchmark Datasets

Table 3 summarizes the results at end-to-end ASR on WSJ0, CHiME3, and TIMIT datasets. Results show that NIESR achieves 5.48%, 6.16%, and 6.61% relative improvements over base model on WSJ0, CHiME3, and TIMIT, respectively, and demonstrates the best CER among all methods.

3.4. Invariance to Nuisance Factors

In order to examine whether a latent embedding is invariant to nuisance factors z , we calculate the accuracy of predicting the factor z from the encoding. Specifically, this is calculated by training classification networks (BLSTM followed by two fully-connected layers) to predict z from the generated embeddings. Table 4 presents results of this experiment, showing that the h^1 embedding of the NIESR model, which is used for ASR, contains less nuisance information than the h encoding of the base, Spk-Inv, and Env-Inv models. In contrast, the h^2 embedding of NIESR contains most of the nuisance information, showing that nuisance factors migrate to this embedding, as expected.

3.5. Additional Robustness through Data Augmentation

Training with additional data that reflects multiple variations of nuisance factors helps models generalize better. In this experiment, we treat the CHiME3 dataset, which contains WSJ0

Table 4: Results of predicting nuisance factor z from learned representations as accuracy. Env stands for environment.

Dataset	Predict z from	Accuracy	
		z : Speaker	z : Env
WSJ0	h in Base Model	67.91	–
	h in Spk-Inv	65.60	–
	h_1 in NIESR	63.35	–
	h_2 in NIESR	97.92	–
CHiME3	h in Base Model	38.52	69.24
	h in Spk-Inv	37.91	69.11
	h in Env-Inv	38.84	66.44
	h_1 in NIESR	35.87	63.45
	h_2 in NIESR	92.28	97.05

Table 5: Test results of models trained on the WSJ0+CHiME3 augmented dataset as CER (%). Values in parentheses show the relative improvement (%) over Base model.

Model	WSJ0	CHiME3
Base	9.35	41.55
Spk-Inv	8.62 (7.81)	40.77 (1.88)
Env-Inv	9.17 (1.93)	40.27 (3.08)
NIESR	8.00 (14.44)	38.35 (7.7)

recordings with four different types of noise, as a noisy augmentation for WSJ0. We train the base model and NIESR on the augmented dataset, i.e. WSJ0+CHiME3, and test on the original CHiME3 and WSJ0 test sets separately. Table 5 summarizes the results on this experiment, showing that training with data augmentation provides improvements on both CHiME3 and WSJ0 datasets compared to the results in Table 3. It is important to note that the NIESR model trained on the augmented dataset achieves 14.44% relative improvement on WSJ0 as compared to the base model trained on the same. This is because data augmentation provides additional information about potential nuisance factors to the NIESR model and, consequently, helps it ignore these factors for the ASR task, even though pairwise data is not provided to the model like [13]. Hence, results show that the NIESR model can be easily combined with data augmentation to further enhance the robustness and nuisance-invariance of the learned features.

4. Conclusion

We presented NIESR, an end-to-end speech recognition model that adopts the unsupervised adversarial invariance framework for invariance to nuisances without requiring any knowledge of potential nuisance factors. The model works by learning a split representation of data through competition between the recognition and an auxiliary data reconstruction task. Results of experimental evaluation demonstrate that the proposed model achieves significant boosts in performance on ASR.

5. Acknowledgements

This material is based on research sponsored by DARPA under agreement number FA8750-18-2-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

6. References

- [1] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition." 2017.
- [2] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.
- [3] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," *arXiv preprint arXiv:1804.10752*, 2018.
- [4] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems*, 2016, pp. 5067–5075.
- [5] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.
- [6] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.
- [7] A. Jaiswal, R. Y. Wu, W. Abd-Almageed, and P. Natarajan, "Unsupervised Adversarial Invariance," in *Advances in Neural Information Processing Systems*, 2018, pp. 5097–5107.
- [8] A. Jaiswal, S. Xia, I. Masi, and W. AbdAlmageed, "RoPAD: Robust Presentation Attack Detection through Unsupervised Adversarial Invariance," in *12th IAPR International Conference on Biometrics (ICB)*, 2019.
- [9] A. Jaiswal, Y. Wu, W. AbdAlmageed, and P. Natarajan, "Unified adversarial invariance," 2019.
- [10] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, "Invariant representations for noisy speech recognition," *arXiv preprint arXiv:1612.01928*, 2016.
- [11] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gang, and B.-H. Juang, "Speaker-invariant training via adversarial learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5969–5973.
- [12] W.-N. Hsu and J. Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5614–5618.
- [13] D. Liang, Z. Huang, and Z. C. Lipton, "Learning noise-invariant representations for robust speech recognition," *arXiv preprint arXiv:1807.06610*, 2018.
- [14] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.
- [15] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4845–4849.
- [16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [17] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.
- [18] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.
- [19] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chimespeech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 504–511.
- [20] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.