Acoustic Feature Learning via Deep Variational Canonical Correlation Analysis

Qingming Tang[1], *Weiran Wang*[1], *Karen Livescu*[1]

[1]Toyota Technological Institute at Chicago, USA
{qmtang,weiranwang,klivescu}@ttic.edu

Abstract

We study the problem of acoustic feature learning in the setting where we have access to another (non-acoustic) modality for feature learning but not at test time. We use deep variational canonical correlation analysis (VCCA), a recently proposed deep generative method for multi-view representation learning. We also extend VCCA with improved latent variable priors and with adversarial learning. Compared to other techniques for multi-view feature learning, VCCA's advantages include an intuitive latent variable interpretation and a variational lower bound objective that can be trained end-to-end efficiently. We compare VCCA and its extensions with previous feature learning methods on the University of Wisconsin X-ray Microbeam Database, and show that VCCA-based feature learning improves over previous methods for speaker-independent phonetic recognition.

Index Terms: multi-view learning, acoustic features, canonical correlation analysis, variational methods, adversarial learning

1. Introduction

Applications involve acoustic speech often benefit from one or more additional types of measurements (e.g. images, video and articulatory measurements) via representing and reasoning about the multiple modalities in the same vector space [1, 2, 3, 4, 5, 6]. A number of approaches have been developed to automatically learn latent spaces that are modality-agnostic. In this work we address the setting where multiple modalities, or "views", are available for feature learning but only one (in this case, acoustics) is available for downstream tasks (e.g., speech recognition). A popular class of methods in this area is based on canonical correlation analysis (CCA, [7]) and its nonlinear extensions [8, 9, 10]. CCA finds projections of two data views that are maximally correlated, subject to uncorrelatedness between learned dimensions in each projected view. This common subspace or projection mappings are often more discriminative or robust to noise, and can be used as good features for downstream tasks.

In this paper, we explore a recently proposed deep generative variant of CCA, deep variational CCA (VCCA) [11], for multi-view acoustic feature learning. VCCA models the joint distribution of two views with a latent variable model capturing both the shared and private (view-specific) information, where the distributions are parameterized by deep neural networks (DNNs). VCCA optimizes a variational lower bound of the likelihood and allows for straightforward training using small minibatches of training samples. VCCA has been found to improve over other multi-view methods for multiple real-world tasks, but thus far not for speech tasks [11]. In this work we study and extend VCCA and show that it can learn acoustic features from parallel acoustic and articulatory data that improve phonetic recognizers.

We compare VCCA and its extensions with previous single-view and multi-view feature learning approaches using the University of Wisconsin X-ray Microbeam Database [12], which consists of simultaneously recorded audio and articulatory measurements. We first show that VCCA learns better acoustic features for speaker-independent phonetic recognition. We then extend VCCA in several ways, including aspects of the prior distributions and adversarial learning [13].

2. Deep variational CCA

Deep CCA (DCCA, [10]), which uses DNNs for projecting two views into a common subspace, has achieved excellent empirical performance for tasks across several domains in the setting of unsupervised multi-view feature learning [14, 15, 16, 17]. However, DCCA has certain drawbacks. First, its objective couples all training samples together and is hard to optimize with stochastic gradient descent (SGD). (Although this is somewhat alleviated by [18], their optimization algorithm requires more tuning parameters.) Second, DCCA focuses on extracting the shared information (most correlated dimensions) only, while there may also be useful view-specific ("private") information that we wish to retain. An autoencoder-based extension to DCCA has been proposed [17], but in practice it produces quite similar results to DCCA. Finally, DCCA (and kernel CCA) are deterministic methods, that is they do not provide a probabilistic model of the data, so they can not generate samples of the input modalities.[1]

Deep variational CCA is based on extending a latent variable formulation of CCA due to Bach and Jordan [19]. Denote the two input views x and y, and assume they are both generated independently from a common (multi-dimensional) latent variable z: $p(x, y, z) = p(z)p(x|z)p(y|z)$. Bach and Jordan [19] showed that, when $p(z)$, $p(x|z)$, and $p(y|z)$ are Gaussian, with the conditional means linear in z, the maximum likelihood solution gives "projections" $\mathbb{E}[z|x]$ and $\mathbb{E}[z|y]$ that lie in the same space as the (deterministic) linear CCA projections. The intuition of a common variable generating multiple views fits our setting, as in our case there is a latent variable, the phonetic transcription, that generates (to a large degree) both acoustic and articulatory measurements and that we wish to recover. However, modeling the complex distribution of speech is too challenging for linear mappings; we therefore consider a deep extension of this generative model.

2.1. Basic variational CCA

VCCA was introduced in [11]; we review the formulation here briefly. In VCCA, the prior $p(z)$ is taken to be a simple Gaussian, but $p_\theta(x|z)$ and $p_\theta(y|z)$ are isotropic Gaussians

[1] This final point is not relevant to our experimental setting in this paper, but can be important more broadly in other settings.

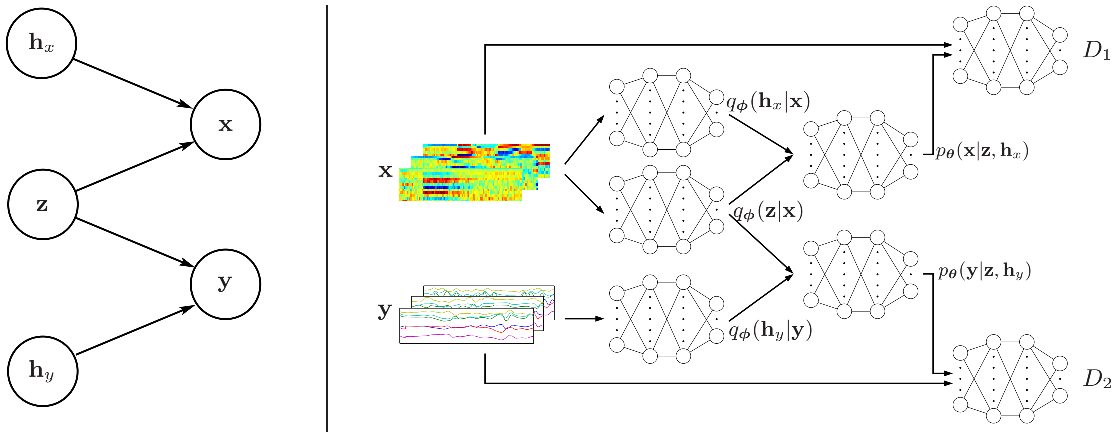
$\mathbf{h}_x$
$\mathbf{x}$
$\mathbf{z}$
$\mathbf{y}$
$\mathbf{h}_y$
D_1
$q_{\phi}(\mathbf{h}_x|\mathbf{x})$
$p_{\theta}(\mathbf{x}|\mathbf{z},\mathbf{h}_x)$
$\mathbf{x}$
$q_{\phi}(\mathbf{z}|\mathbf{x})$
$p_{\theta}(\mathbf{y}|\mathbf{z},\mathbf{h}_y)$
$\mathbf{y}$
$q_{\phi}(\mathbf{h}_y|\mathbf{y})$
D_2

Figure 1: *Left: graphical model of VCCA with view-specific private variables (VCCAP). Right: multi-view acoustic feature learning with VCCAP and adversarial training. VCCAP alone corresponds to this model without D1, D2; basic VCCA also has no h_x, h_y.*

with means parameterized by two DNNs respectively (parameters collectively denoted θ) and whose standard deviations are a tuning parameter. With this parameterization, the marginal distribution $p(x,y) = \int p(z)p(x|z)p(y|z)dz$ does not have a closed form. Using a variational inference approach similar to that of variational autoencoders [20], VCCA parameterizes the approximate posterior $q_\phi(z|x)$ using another DNN (with weight parameters ϕ), and optimizes the following lower bound of the log likelihood :

$$\mathcal{L}(x,y;\theta,\phi) = -D_{KL}(q_\phi(z|x)||p(z)) \\ + E_{q_\phi(z|x)}[\log p_\theta(x|z) + \log p_\theta(y|z)]. \quad (1)$$

Here $D_{KL}(q_\phi(z|x)||p(z))$ denotes the KL divergence between the learned $q_\phi(z|x)$ and the prior $p(z)$. Note that, for Gaussian observation models $p(x|z)$ and $p(y|z)$, maximizing the likelihood is equivalent to minimizing the reconstruction errors. Calculating the expectation term in 1 can be done efficiently by Monte Carlo sampling. The approximate posterior is modeled as another Gaussian whose mean and standard deviation are outputs of the neural network with weights ϕ. Using the "reparameterization trick" of [20], we only need to draw samples from the simple $\mathcal{N}(0,I)$ distribution to obtain samples of $q_\phi(z|x)$, and stochastic gradients with respect to the DNN weights can be computed using standard backpropagation.

Given a set of N paired samples $\{(x_1,y_1),...,(x_N,y_N)\}$ of the input variables (x,y), VCCA maximizes the variational lower bound on the training set

$$\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x_i,y_i;\theta,\phi).$$

Finally, as in the probabilistic interpretation of CCA, we use the mean of $q_\phi(z|x)$ (the outputs of the corresponding DNN when fed x as input) as the learned features.

2.2. VCCA with private variables (VCCAP)

VCCA has been extended by [11] to include view-specific private variables; we denote the resulting model VCCA-private (VCCAP). VCCAP introduces two additional sets of latent variables, h_x of dimensionality d_{h_x} and h_y of dimensionality d_{h_y}, for the two views x and y respectively. The input x is taken to be jointly generated by h_x and z, and similarly for the second view. The intuition is that there might be aspects of the two views that are not captured by the shared variables, so that reconstructing the inputs using only the shared variables is difficult. For example, in the case of acoustic and articulatory data used in our experiments, the articulatory data does not contain nasality information while the acoustic signal does. Explicitly modeling the private information in each view eases the burden of the reconstruction mappings, and may also retain additional useful information in the target view. The graphical model of VCCAP is shown in Figure 1 (left).

To obtain a tractable objective, VCCAP parameterizes two additional posteriors $q_\phi(h_x|x)$ and $q_\phi(h_y|y)$ and maximizes the following lower bound on the log likelihood:

$$\begin{aligned} \mathcal{L}_{private}(x,y;\theta,\phi) &= -D_{KL}(q_\phi(z|x)||p(z)) \\ -D_{KL}(q_\phi(h_y|y)||p(h_y)) &- D_{KL}(q_\phi(h_x|x)||p(h_x)) \\ &+E_{q_\phi(z|x)q_\phi(h_x|x)}[\log p_\theta(x|z,h_x)] \\ &+E_{q_\phi(z|x)q_\phi(h_y|y)}[\log p_\theta(y|z,h_y)] \quad (2) \end{aligned}$$

Here we use simple $\mathcal{N}(0,I)$ priors for h_x and h_y. Given a training set, we maximize this variational lower bound on the training samples using SGD and Monte Carlo sampling as before. Note that, unlike DCCA, the VCCA and VCCAP objectives are sums over training samples, so it is trivial to use SGD with small minibatches. Again, we use the mean of $q_\phi(z|x)$ as the learned features for VCCAP.

3. Extensions of VCCA

We now propose two extensions of VCCA/VCCAP, which will be explored later in our experiments.

3.1. Tuning the KL divergence terms

The KL divergence terms in the objectives of VCCA and VCCA-private naturally arise in the variational lower bound derivation. On the other hand, they can also be understood as regularization terms that control the complexity of the projection networks $q_\phi(z|x)$, $q_\phi(h_x|x)$ and $q_\phi(h_y|y)$: They enforce our belief that the posterior distribution should be close to $\mathcal{N}(0,I)$ so that each dimension has approximately zero mean and unit scale, and different dimensions are independent.

Taking this regularization view, we can further control the regularization effect in two ways. First, we can add a weight β on the KL divergence terms ($\beta = 1$ for standard

VCCA/VCCAP). Second, we could add more informative priors than $\mathcal{N}(0, I)$. For our ASR task, we typically concatenate input frames over a context window. We find that features learned at a moderate window size can be used as priors for learning at a larger context window size. Such priors may both improve the feature quality and speed up training over using the simple Gaussian prior. Experiments demonstrating this effect are given in Section 4.3.1.

3.2. Generative adversarial training

Adversarial learning is an increasingly popular approach for learning deep generative models [13, 21, 22, 23, 24]. In this approach there are two modules: a discriminator that takes a data sample and tries determine whether it is generated by a model (fake) or comes from the real data distribution (e.g., a sample from the training set), and a generator that produces data samples from the generative model and tries to fool the discriminator. At equilibrium, generative adversarial networks (GANs) should be able to capture well the data distribution.

We extend VCCA/VCCAP with generative adversarial training by viewing the basic VCCA/VCCAP model as a pair of generators, and adding two discriminators D_1 and D_2, one for each view, which try to distinguish the generated samples from training set input samples. Accordingly, we augment the original VCCA objective with losses on the discriminators. For fixed generators, D_1 optimizes the following objective for the i-th sample (D_2 maximizes an analogous objective):

$$\max_{D_1} \; \log D_1(x_i) + \log(1 - D_1(x_i')) \tag{3}$$

where x_i is the original input of the first view and x_i' is the corresponding reconstruction. Given the discriminators, the generators (including the projection networks and reconstruction networks) optimize the following objective for the i-th sample

$$\max_{\theta,\phi} \; \mathcal{L}_{private}(x_i, y_i; \theta, \phi)$$
$$+\lambda_1 \log(D_1(x_i')) + \lambda_2 \log(D_2(y_i')).$$

Here λ_1 and λ_2 are two hyperparameters used to trade off the GAN loss with the VCCA/VCCAP loss. Note that x_i' and y_i' are functions of the generators (and thus also parameterized by θ, ϕ), and the generators try to make D_1 and D_2 accept fake samples (reconstructions). We alternate the above two stages during training. The model architecture incorporating GANs is depicted in Figure 1 (right). Experiments demonstrating this approach are given in Section 4.3.2.

4. Experimental Results

The Wisconsin X-ray microbeam (XRMB) [12] corpus consists of both speech and articulatory measurements, which are simultaneously recorded, from 47 American English speakers. Previous works [5, 14] have shown that features learned by linear, kernel and deep CCA can be used to improve performance of speaker-independent phonetic recognition on this corpus. Note that the recognizer is trained and tested using acoustic view only though features are learned using both views.

Following the basic setup of [14], the two input views are based on standard $39D$ MFCC+$\Delta+\Delta\Delta$s and $16D$ articulatory features (horizontal/vertical displacement of 8 pellets attached to several parts of the vocal tract). To incorporate context information, the inputs are concatenated over a W-frame window centered at each frame, giving $39 \times W$ and $16 \times W$ feature dimensions for each of the two views respectively.

The 47 speakers of XRMB are divided into disjoint sets of $35/12$ speakers for feature learning and recognition respectively. The 12 recognition speakers are further divided into 6 disjoint groups to perform a 6-fold experiment. In each fold, $8/2/2$ speakers are used for training/tuning/testing the recognizer respectively. For each fold, we choose the hyperparameters that give the best PER on the fold-specific dev set and measure its corresponding test result; finally, we report the average test PER over the 6 folds. Each split/fold is gender-balanced, and the per-speaker mean and variance of the articulatory measurements are removed for the feature learning stage. Unlike [14], we include silence frames in feature learning, yielding a total of 1.7M training samples. (Although the added frames correspond to silence, the context window often overlaps speech.) This larger training set benefits the generative VCCA/VCCAP methods.

4.1. Algorithms and architectures

We compare VCCA/VCCAP and its extensions with baseline MFCCs, multi-view features learned with deep CCA (DCCA [10, 14]) and multi-view contrastive loss (Contrastive [25]). The contrastive loss aims to make the distance between paired samples smaller than the distance between unpaired (negative) samples by a margin. Note that in previous work [14], DCCA was already found to outperform supervised DNN bottleneck features (trained on the recognizer training data and used in a tandem recognizer [26]) for $W = 7$. Both DCCA and Contrastive were compared against VCCA/VCCAP in [11] for a 7-frame context window, where VCCAP was slightly outperformed by Contrastive when silence frames are not used in training. In this paper, we show that VCCAP and its extensions outperform these alternatives when all methods use the same larger training set at different context sizes.

DNN architectures and training procedures for DCCA and Contrastive used here are consistent with previous work [11], and the VCCA-based methods use the same architectures. Each DNN has up to 3 ReLU [27] hidden layers: each hidden layer of $q(z|x)$ has 1500 units while each hidden layer of $q(h_x|x)$ and $q(h_y|y)$ has 1024 units, and the discriminators D_1 and D_2 use $[2048, 1500, 1500]$ units in each hidden layer. With limited tuning at context size $W = 7$, we fix the standard deviations of $p_\theta(x|z, h_x)$ and $p_\theta(y|z, h_y)$ to 1 and 0.1 respectively. Our methods are trained using the Adam optimizer [28] with a fixed learning rate of 0.0001 and minibatch size 200, except that the discriminators of VCCAP+GAN are updated less frequently using minibatches of size 1800. Dropout [29] at a rate of 0.2 is used for all layers. The overall best performing VCCA, VCCAP and DCCA models have a feature dimensionality of 70, while the best feature dimensionality is 50 for Contrastive. We train the four models to convergence (around 60, 300, 60 and 30 epochs for VCCA, VCCAP, DCCA and Contrastive loss, respectively).

We use two phone recognizers in this paper. The first is a 3-state monophone HMM/GMM system, which is fast and consistent with previous work [14]. This recognizer is built with Kaldi [30] and uses a TIMIT bigram language model. The learned features are used in a tandem approach [26] for this recognizer, i.e., they are concatenated with the original $39D$ MFCCs and used as input to the HMM/GMM recognizer. The second recognizer is a connectionist temporal classification (CTC)-based recognizer [31], built with TensorFlow [32], that uses a 2-layer bidirectional recurrent neural network (RNN) with 512 gated recurrent unit (GRU [33]) cells in each layer.

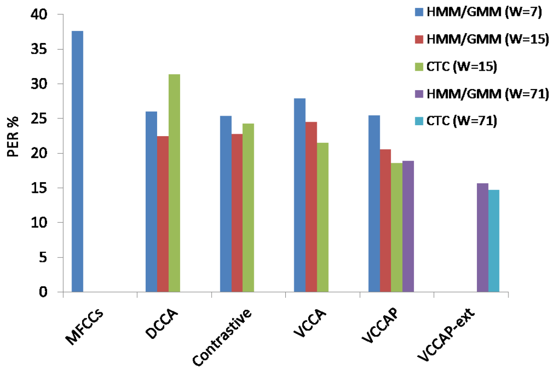

Figure 2: *Average test set phone error rate (PER, %) over 6 folds obtained by HMM/GMM and CTC-based recognizers with features learned by different methods at window size $W = 7/15/71$. VCCAP-ext is the extension of VCCAP with a weighted KL divergence term and learned priors.*

W	3	7	15	35	41	61	71
PER	27.4	22.8	19.5	15.1	15.6	14.9	16.1

Table 1: *Development set PER (%) of one fold obtained with the HMM/GMM recognizer and VCCAP features for different input context sizes when trained for* 60 *epochs.*

We do not concatenate features with the original MFCCs as this tends to give better performance on dev sets. No language model is used for this recognizer. We decode with beam search with a beam width of 100. Since experiments with the CTC-based recognizer are much slower than with the HMM/GMM, we provide CTC results for a subset of the experiments.

4.2. Main test set results and effect of window size

We first give the main final test set results, summarized in Figure 2, and then describe additional comparisons on the dev set.

We observe from Figure 2 that all deep multi-view features benefit from wider context, and significantly improve over the original MFCC features. Using either HMM/GMM or CTC-based recognizers, VCCAP outperforms basic VCCA, is comparable to Contrastive and DCCA for $W = 7$, and outperforms all alternative methods for $W = 15$. We therefore focus on VCCAP in the remaining experiments and discussion. The best performance is obtained with the extension to VCCAP proposed in Section 3.1, with a weight on the KL divergence term (of $\beta = 10$) and where the $\mathcal{N}(0, I)$ prior for a large-window input (here $W = 71$) is replaced by the posterior learned with a smaller window (here $W = 35$).

Table 1 provides a more detailed look at the role of the context window size, in terms of development set PER in one fold (to avoid excessive test set reuse). When fixing the other hyperparameters of VCCAP, the PER initially decreases quickly with increasing context window size, then plateaus after $W = 15$, and starts to increase again at 71 frames. It is only by using the learned prior extension that we are able to obtain further improvements with larger context windows.

4.3. VCCAP Extensions

We next provide a more detailed study of the proposed VCCAP extensions. To avoid excessive test set reuse and to speed up experimentation, we report dev set results from a single fold when each model is trained for 60 epochs. We then use the learned features only in the HMM/GMM recognizer.

	$W = 35$	51	61	71
$\beta = 1, \mathcal{N}(0, I)$ prior	15.1	14.9	14.9	16.1
$\beta = 10, \mathcal{N}(0, I)$ prior	15.1	14.2	15.1	14.7
$\beta = 10$, learned prior	-	14.4	13.8	13.2

Table 2: *Development set PER (%) obtained with the HMM/GMM recognizer and VCCAP features for different KL divergence term penalty weights, with/without priors learned by the 35-frame window model.*

4.3.1. Tuning the KL divergence terms

Table 1 shows that learning from very large context windows is more challenging. Table 2 shows that the proposed VCCAP extensions of Section 3.1 allow for further gains with larger windows. Specifically, with a very large window of $W = 71$, giving more weight to the KL divergence term improves performance, and using a prior corresponding to the learned posterior from the $W = 35$ model helps further. These extensions force the model to pay more attention on the central 35 frames.

4.3.2. Incoporating adversarial learning

Finally, we experiment with the VCCAP+GAN approach (Section 3.2). We train VCCAP with the best hyperparameters for 30 epochs; then starting from this model we train VCCAP+GAN and VCCAP both for another 30 epochs each. Tradeoff parameters for the discriminators ($\lambda_1 = \lambda_2 = 5$) are tuned at $W = 7$ based on average dev set PER. VCCAP+GAN slightly outperforms VCCAP, with gains of 0.1-0.4 in dev set PER across folds for $W = 7/15/35$, but the improvement is less stable than that of the other extensions. Considering the added complexity of GAN training, the results may not justify adopting this approach in general, and for our final test results we did not include this extension.

5. Conclusion

We have shown that VCCA-private and its extensions improve over previous approaches for unsupervised acoustic feature learning from multi-view acoustic and articulatory data, and in particular that the method benefits from large input context windows using learned priors. In fact, we have observed that we can continue to improve performance with even larger windows by re-learning priors, but thorough experimentation becomes impractical; this suggests that recurrent models may be fruitful, as a way of handling very large context. Further avenues for research in unsupervised speech processing include using the extracted private variables (which are unused in our work thus far), which may be useful for generation tasks like voice morphing and other synthetic speech generation. The setting could also be extended to semi-supervised learning, that is joint training of the recognizer and VCCA objectives, as well as for learning sequence-level features.

6. Acknowledgments

This research was supported by NSF grant IIS-1321015 and used GPUs donated by NVIDIA Corporation.

7. References

[1] T. J. Hazen, B. Sherry, and M. Adler, "Speech-based annotation and retrieval of digital photographs," in *Proc. Interspeech*, 2007.

[2] G. Chrupała, L. Gelderloos, and A. Alishahi, "Representations of language in a model of visually grounded speech signal," *arXiv preprint arXiv:1702.01991*, 2017.

[3] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[4] J. Huang and B. Kingsbury, "Audio-visual deep learning for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, 2013.

[5] R. Arora and K. Livescu, "Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains," in *Proc. IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, 2013.

[6] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "Integrating articulatory data in deep neural network-based acoustic modeling," *Computer Speech & Language*, vol. 36, pp. 173–195, 2016.

[7] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[8] T. Melzer, M. Reiter, and H. Bischof, "Nonlinear feature extraction using generalized canonical correlation analysis," in *Proc. Int. Conf. on Artificial Neural Networks*, 2001.

[9] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, no. Jul, pp. 1–48, 2002.

[10] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis." in *Proc. ICML*, 2013.

[11] W. Wang, X. Yan, H. Lee, and K. Livescu, "Deep variational canonical correlation analysis," *arXiv preprint arXiv:1610.03454*, 2016.

[12] J. Westbury, P. Milenkovic, G. Weismer, and R. Kent, "X-ray microbeam speech production database," *Journal of the Acoustical Society of America*, vol. 88, no. S1, pp. S56–S56, 1990.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[14] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, 2015.

[15] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu, "Deep multilingual correlation for improved word embeddings." in *Proc. Human Language Technology/Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, 2015.

[17] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *Proc. ICML*, 2015.

[18] W. Wang, R. Arora, N. Srebro, and K. Livescu, "Stochastic optimization for deep CCA via nonlinear orthogonal iterations," in *53nd Annual Allerton Conference on Communication, Control and Computing*, 2015.

[19] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," U. C. Berkeley, Tech. Rep., 2005.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Proc. Int. Conf. Learning Representations*, 2014.

[21] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *Proc. Int. Conf. Learning Representations*, 2017.

[22] M. Chen and L. Denoyer, "Multi-view generative adversarial networks," *arXiv preprint arXiv:1611.02019*, 2016.

[23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017.

[24] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *Proc. Int. Conf. Learning Representations*, 2017.

[25] K. M. Hermann and P. Blunsom, "Multilingual models for compositional distributed semantics," *Proc. Association for Computational Linguistics (ACL)*, 2014.

[26] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Proc. IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, vol. 3, 2000.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, 2010.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. Learning Representations*, 2015.

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[31] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.